

VISIT...

LANZAROTE
Caliente.COM

Д.Б. Юдин, Е.Г. Гольштейн

ЗАДАЧИ И МЕТОДЫ
ЛИНЕЙНОГО
ПРОГРАММИРОВАНИЯ

МОСКОВСКИЙ РАЙОН

ЮДИН Д. Б., ГОЛЬШТЕЙН Е. Г.

ЗАДАЧИ И МЕТОДЫ
ЛИНЕЙНОГО
ПРОГРАММИРОВАНИЯ

ИЗДАТЕЛЬСТВО „СОВЕТСКОЕ РАДИО“

МОСКВА — 1961

Книга является первым в отечественной литературе систематическим изложением теоретических основ, методов и приложений новой математической дисциплины — линейного программирования. Основное внимание здесь обращено на обоснование и описание вычислительных алгоритмов, которые доводятся до расчетных схем и иллюстрируются примерами.

Книга предназначена для широкого круга специалистов — математиков, инженеров и экономистов с повышенной математической подготовкой.

ПРЕДИСЛОВИЕ

В последние десятилетия значительно расширились области приложения математики. Это, в свою очередь, обусловило появление новых методов исследований. Одной из новых математических дисциплин, вызванных к жизни потребностями рационального планирования производства, является линейное программирование. В настоящее время приложения линейного программирования вышли за рамки экономических задач. Методы этой дисциплины находят широкое применение в задачах, связанных с управлением различными процессами, с техническими расчетами, с рациональным использованием военной техники. Наметились существенные связи линейного программирования с другими математическими дисциплинами.

Линейное программирование во многих случаях дает возможность установить, как следует распорядиться имеющимися ресурсами для достижения наилучшего результата. Подобные задачи возникают у экономистов в связи со сложными вопросами планирования народного хозяйства, у инженеров при разработке методов управления техникой, у математиков при решении некоторых экстремальных задач.

Несмотря на то, что линейное программирование возникло в Советском Союзе, до сих пор в отечественной литературе нет систематического описания задач и методов этой дисциплины, которое могло бы служить учебным пособием,

Авторы поставили перед собой задачу подробно изложить основные методы линейного программирования в форме, доступной для инженеров с несколько повышенной математической подготовкой. Книга предназначена главным образом для тех, кто непосредственно участвует в постановке и решении задач планирования и управления. В связи с этим в ней уделено много внимания доведению методов до четких вычислительных алгоритмов.

Методы линейного программирования относительно универсальны. Тем не менее учет особенностей конкретной задачи обычно приводит к сокращению трудоемкости вычислений. Применение линейного программирования для решения практических задач — процесс творческий. Непосредственному решению задачи, как правило, должен предшествовать анализ, позволяющий выявить особенности задачи и определить наиболее рациональный метод ее исследования. После этого целесообразно, используя специфику задачи, упростить алгоритм выбранного метода. Таким образом, строгое обоснование методов не менее важно, чем четкое изложение алгоритмов.

В первой главе книги излагаются основные понятия и различные интерпретации задачи линейного программирования. Параграфы 1—4 этой главы носят элементарный иллюстративный характер. Последующее изложение более строгое. При первом чтении можно опустить § 9 и вернуться к нему после изучения общих методов линейного программирования.

Во второй главе описаны различные практические задачи, приводящие к общей и частным проблемам линейного программирования. Основная цель главы — иллюстрация приемов, позволяющих формулировать экономические, технические, военные и другие задачи в терминах линейного программирования.

Третья и четвертая главы посвящены общим методам — их обоснованию и соответствующим алгоритмам. Подробней других описан метод последовательного улучшения плана (гл. 3). Изучение главы 3 не требует специальной подготовки. Исключение представляет § 7, который может быть опущен при первом чтении. Описание двойственных методов (гл. 4) проведено в более сжатой форме.

В главе 5 подробно изучается одна из важных частных проблем линейного программирования — транспортная задача. Читателю, заинтересованному лишь в вычислительных алгоритмах, достаточно познакомиться с § 2 и 3 этой главы. Материал главы 5 представляет самостоятельный интерес. Поэтому авторы сочли целесообразным не связывать изложение методов решения транспортной задачи с общими методами линейного программирования. Читатель, интересующийся только проблемой перевозок, может изучать главу 5, не обращаясь к предыдущим главам. Доказательства некоторых утверждений главы 5 вынесены в § 3 главы 6.

В пункте 1.1 главы 6 определяются и поясняются основные понятия многомерного пространства, использованные в книге. Остальная часть шестой главы содержит математические основы, необходимые для более углубленного изучения теории линейного программирования. Этот материал представляет интерес главным образом для лиц, работающих над усовершенствованием методов и вычислительных алгоритмов.

Наконец, в заключении кратко излагаются специальные вопросы линейного программирования, которые по различным причинам не включены в книгу.

В списке литературы, помещенном в конце книги, перечислены лишь источники, в той или иной мере использованные в тексте.

Описание всех методов линейного программирования ведется в книге по единой схеме. Вначале приводятся

теоретические основы метода, затем излагается алгоритм, реализующий метод, и подробно описывается решение численного примера. Усвоение вычислительного алгоритма не требует предварительного ознакомления с обоснованием метода. Некоторые навыки решения задач линейного программирования читатель может получить, детально ознакомившись с одними лишь численными примерами. Авторам представляется, что принятая структура глав позволит специалистам разного профиля и различной математической подготовки выбрать удобный для себя порядок изучения линейного программирования. Следует, однако, еще раз подчеркнуть, что для эффективного использования приведенных методов необходимо знать как вычислительные, так и теоретические аспекты линейного программирования.

Настоящая книга представляет собой первый опыт систематического изложения задач и методов линейного программирования. По-видимому, она не лишена недостатков. Замечания и советы по их устранению будут приняты с благодарностью.

При редактировании книги весьма полезными оказались советы проф. Ляпунова А. А., Полетаева И. А., Гурина Л. С., Мовшовича С. М. и Боковой В. В. Всем этим лицам авторы выражают свою искреннюю признательность.

Ноябрь 1959 г.

Авторы

ГЛАВА 1

ОСНОВНЫЕ ПОНЯТИЯ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

§ 1. ПРЕДМЕТ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

1.1. Одной из характерных особенностей современной науки является широкое проникновение результатов и методов математики в самые разнообразные области исследования. Сложные задачи управления народным хозяйством и автоматизации производственных процессов, проблемы планирования экономики и военных операций требуют наряду с изучением качественных закономерностей явлений тщательного исследования их количественных характеристик

Математические машины, внедряемые в производство и управление и используемые в научно-исследовательской работе, создают огромные возможности для развития различных отраслей науки, для совершенствования методов планирования и автоматизации производства. Однако без строгих формулировок задач, без формально-математического описания процессов не может быть достигнут необходимый уровень использования техники. Зачастую «узким местом» оказываются не столько вопросы, связанные с проектированием машин, сколько вопросы, возникающие в связи с формализацией физических, экономических, технических и других процессов. Формализация задачи — необходимый этап для перевода каждой прикладной задачи на язык математических машин.

Общая схема формализации задач управления и планирования может быть описана следующим образом,

Задачи управления и планирования обычно сводятся к выбору некоторой системы параметров или некоторых систем функций. Чтобы иметь основание отдавать предпочтение тем или иным значениям параметров планирования и тем или иным управляющим функциям, необходимо прежде всего четко уяснить себе следующие два обстоятельства. Во-первых, нужно разобраться в том, что следует называть хорошей и наилучшей системой управления или планирования. Иными словами, следует сформулировать и выразить через искомые характеристики показатель качества — критерий, определяющий соответствие разрабатываемых устройств и планов цели, ради которой эта разработка ведется. Во-вторых, необходимо выяснить условия работы системы и вытекающие отсюда ограничения, которым должны удовлетворять искомые параметры и характеристики.

1.2. В весьма широком классе народно-хозяйственных, технических, военных и других задач показатель качества выражается линейно через параметры управления или характеристики планирования, а условия, которым должны удовлетворять искомые параметры, записываются в виде линейных равенств или неравенств. Вычисление экстремума (максимума или минимума) линейного показателя качества при условии, что переменные, подлежащие определению, удовлетворяют линейным ограничениям, составляет предмет линейного программирования. Актуальность подобных задач в народном хозяйстве и в военном деле в последние годы вызвала к жизни большое количество статей, посвященных различным методам определения экстремума линейных форм и применению этих методов к решению практических задач.

Термин «линейное программирование» нельзя признать удачным, поскольку под словом «программирование» обычно понимают теорию составления программ для универсальных математических машин. Возможно, более удачным названием совокупности вопросов, оформляющейся в настоящее время в самостоятельный раздел прикладной математики, было бы «линейное планирование». Однако термин «линейное программирование» уже установился и является общепринятым в литературе, поэтому мы его будем придерживаться. Заметим лишь, что слово «программирование» употребляется здесь

в смысле методов (алгоритмов) для решения экстремальных задач, а эпитет «линейное» подчеркивает линейность исследуемой функции и ограничений, наложенных на ее переменные.

1.3. Общая задача линейного программирования формулируется следующим образом.

Требуется найти максимум (или минимум) линейной функции n переменных $x_1, x_2, \dots, x_n$:

$$L = L(x_1, x_2, \dots, x_n) = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \quad (1.1)$$

при ограничениях, наложенных на переменные $x_1, x_2, \dots, x_n$, вида

[illegible]

[illegible]

$$x_1 \geq 0, \dots, x_t \geq 0 \quad (t \leq n). \quad (1.4)$$

Линейную функцию (1.1) — критерий качества выбранных переменных — принято называть линейной формой задачи, а множество наборов $(x_1, x_2, \dots, x_n)$, удовлетворяющих условиям (1.2) — (1.4), — областью определения задачи линейного программирования или областью определения линейной формы.

На первый взгляд может показаться, что сформулированная задача (1.1) — (1.4) весьма проста и легко решается обычными методами классического анализа. Однако это не так. Известные методы решения задач на условный экстремум с помощью так называемых множителей Лагранжа здесь неприменимы. Линейность функции $L(x_1, \dots, x_n)$ вносит в экстремальную задачу определенную специфику. Дело в том, что линейная функция, заданная в ограниченной области, достигает своего максимума и минимума на границе области в точках, в которых соответствующие частные производные, вообще говоря, не обращаются в нуль. Это значит, что

классические методы дифференциального исчисления, разработанные для отыскания экстремальных точек внутри области определения функции, здесь не могут быть использованы.

При анализе системы условий (1.2)—(1.4) могут представиться три случая:

а) Условия (1.2)—(1.4) противоречивы. Другими словами, не существует набора чисел $x_1, \dots, x_n$, удовлетворяющих всем условиям задачи.

б) Условия (1.2)—(1.4) непротиворечивы, а область, определяемая ими, неограничена. Другими словами, существуют наборы чисел $x_1, \dots, x_n$, удовлетворяющие системе (1.2)—(1.4) и содержащие отдельные переменные со сколь угодно большими значениями.

в) Система условий (1.2)—(1.4) совместна и область, определяемая ею, ограничена.

Практические задачи обычно приводят к последнему случаю. Дальше будет показано, что в этом случае область, заданная системой (1.2)—(1.4), представляет собой выпуклый многогранник в n -мерном пространстве и экстремум линейной формы L достигается в его вершинах. Однако уже при относительно небольших n , r и t (случаи, к которым сводятся сравнительно простые практические задачи), количество вершин многогранника исчисляется многими миллиардами. Это значит, что неупорядоченный перебор вершин с целью выявления точки, в которой линейная форма достигает своего наибольшего или наименьшего значения, по-видимому, практически невыполнимая работа даже для самых быстродействующих современных вычислительных машин. Как мы увидим дальше, различные методы решения задач линейного программирования представляют собой те или иные рекомендации по упорядочению перебора вершин многогранника, определяемого условиями задачи.

Введем понятие разрешимости задачи, важное для последующего изложения.

Задача линейного программирования называется разрешимой, если существует набор чисел $(x_1, x_2, \dots, x_n)$ ($x_i < \infty$), удовлетворяющий всем ограничениям (1.2)—(1.4) и обращающий линейную форму в максимум или минимум (в зависимости от постановки задачи),

Неразрешимость задачи может быть обусловлена либо противоречивыми условиями задачи (случай а), либо неограниченностью области определения линейной формы (случай б). В последнем случае неразрешимость задачи связана с неограниченностью линейной формы в области ее определения. Заметим, что случай б) не обязательно приводит к неразрешимой задаче. Линейная форма может быть ограничена и в неограниченной области. В этом случае введение дополнительного условия приводит область определения задачи к выпуклому многограннику (§ 5). Таким образом, во всех случаях, когда задача разрешима, решение ее сводится к упорядоченному перебору вершин некоторого многогранника.

§ 2. ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Во второй главе будут подробно разобраны практические задачи, укладывающиеся в модель (1.1)—(1.4). Здесь мы намерены лишь кратко ознакомить читателя с приложениями линейного программирования.

В п. 2.1 в задаче об организации снабжения выделены характерные случаи, в которых решение задачи линейного программирования не требует специальных методов и громоздких вычислений.

В п. 2.2 приведена постановка задачи о рациональном выборе системы вооружения. Перевод критерия качества и ограничений на формальный язык здесь сразу приводит к задаче вида (1.1)—(1.4). В отличие от этого задача о размещении заказов и загрузке оборудования, описанная в п. 2.3, сводится к задаче линейного программирования только после некоторых (правда несложных) искусственных построений. Другая особенность примера из п. 2.3 заключается в том, что переменные здесь зависят от двух индексов.

2.1. Организация снабжения. Рассмотрим задачу о рациональной организации снабжения промышленного центра однородным продуктом, например картофелем или углем. Продукт может привозиться в центр из n пунктов.

Пусть x_i — количество продукта, поставляемое в центр из i -го пункта, а c_i — суммарная стоимость производства и перевозки единицы продукта из i -го пункта. Тогда стоимость продукта, доставленного из i -го

пункта в центр, равна $c_i x_i$, а стоимость продукта, привезенного в центр из всех пунктов производства, равна

$$L = c_1 x_1 + c_2 x_2 + \dots + c_n x_n. \quad (1.5)$$

Требуется организовать снабжение (выбрать x_i) таким образом, чтобы обеспечить минимальную стоимость продукта в центре. При этом необходимо учесть следующие условия: потребность центра в продукте определяется величиной b ; она должна быть удовлетворена; излишков не должно быть. Следовательно, искомые переменные должны подчиняться условию.

$$x_1 + x_2 + \dots + x_n = b. \quad (1.6)$$

Производство продукта в i -м пункте ограничено величиной b_i , а пропускная способность транспорта, который может быть использован для перевозки продукта из i -го пункта, ограничена величиной d_i .

Обозначим через β_i меньшее из двух чисел b_i и d_i

$$\beta_i = \min(b_i, d_i), \quad i = 1, 2, \dots, n.$$

Искомые переменные x_i должны, таким образом, удовлетворять ограничениям

$$x_i \leq \beta_i, \quad i = 1, 2, \dots, n. \quad (1.7)$$

Кроме того, значения x_i должны быть неотрицательны, так как перевозки продукта из центра в пункты производства этого продукта исключены

$$x_i \geq 0, \quad i = 1, 2, \dots, n. \quad (1.8)$$

Если перевозки из i -го пункта могут быть обеспечены двумя видами транспорта, то i -й пункт целесообразно рассматривать как два пункта с разной стоимостью продукта.

Таким образом, мы пришли к задаче линейного программирования. Необходимо обратить в минимум линейную форму L (1.5) при линейных условиях (1.6) — (1.8).

Сформулированная задача обладает особенностью, позволяющей без труда вычислить значения x_i , при которых будет обеспечена минимальная стоимость продукта в центре.

Очевидно, выгодно получать возможно больше продукта из тех пунктов, для которых величина c_i (стоимость производства и перевозки единицы продукта) мала. Перенумеруем пункты производства в порядке возрастания коэффициентов c_i , т. е. предположим, что

$$c_1 \leq c_2 \leq \dots \leq c_n.$$

По условию из первого пункта (откуда поставляется наиболее дешевый продукт) центр может получить не более β_1 единиц продукта. Если общая потребность центра, измеряемая величиной b , не превышает возможностей первого пункта, т. е. $b \leq \beta_1$, то рациональней всего удовлетворить весь спрос только за счет первого пункта. В этом случае

$$x_1 = b, \quad x_2 = x_3 = \dots = x_n = 0.$$

Если же $b > \beta_1$, то целесообразно обеспечить максимально возможную поставку из первого пункта ($x_1 = \beta_1$), а оставшуюся часть потребностей ($b - \beta_1$) удовлетворить за счет других пунктов производства. Мы пришли, таким образом, к задаче организации снабжения, аналогичной предыдущей, с тем лишь различием, что потребность центра определяется теперь величиной $b - \beta_1$, а число пунктов производства равно $n - 1$. Здесь снова могут представиться два случая. В первом случае $b - \beta_1 \leq \beta_2$. Тогда рациональная система снабжения определяется величинами

$$x_2 = b - \beta_1, \quad x_3 = x_4 = \dots = x_n = 0$$

($x_1 = \beta_1$ определено ранее).

Во втором случае, когда $b - \beta_1 > \beta_2$, целесообразно положить $x_2 = \beta_2$ и перейти к новой задаче снабжения с меньшим числом пунктов производства ($n - 2$) и меньшими потребностями центра ($b - \beta_1 - \beta_2$).

Приведенные рассуждения подсказывают следующий путь решения задачи.

Из величины b последовательно вычитаются числа $\beta_1, \beta_2, \dots$. Могут встретиться два случая:

- а) $b - \beta_1 - \beta_2 - \dots - \beta_n > 0$;
- б) $b - \beta_1 - \beta_2 - \dots - \beta_n \leq 0$.

В первом случае невозможно полностью удовлетворить потребности центра. Спрос превышает суммарную возможность поставок продукта из всех пунктов производства.

Во втором случае спрос может быть полностью удовлетворен.

Определим индекс k из условия

$$b - \beta_1 - \dots - \beta_k \geq 0,$$

$$b - \beta_1 - \dots - \beta_k - \beta_{k+1} < 0.$$

Тогда рациональная система снабжения определяется следующими соотношениями

$$x_1 = \beta_1, x_2 = \beta_2, \dots, x_k = \beta_k;$$

$$x_{k+1} = b - \beta_1 - \beta_2 - \dots - \beta_k; \quad x_{k+2} = \dots = x_n = 0.$$

Таким образом, в оптимальном плане снабжения возможности первых k пунктов используются полностью, а последние $n - (k+1)$ пунктов в план поставок не включаются вовсе.

Как видим, для решения рассмотренной задачи потребовались лишь элементарные соображения. К сожалению, это не правило, а скорее исключение в задачах линейного программирования. Простота решения здесь является следствием того, что условия задачи содержат только одно ограничение, связывающее все переменные. Все остальные условия ограничивают только область изменения каждой из переменных.

Усложним теперь задачу о снабжении промышленного центра однородным продуктом, включив в нее дополнительное ограничение.

Пусть, например, время, в течение которого транспорт, предназначенный для перевозки продукта, может находиться под погрузкой, ограничено, а механизация погрузочных работ в разных пунктах производства различна. Обозначим через a_i время, затрачиваемое в i -м пункте на погрузку одной транспортной единицы, а через T — ограничение по сумме времен, затрачиваемых всем транспортом на погрузочные работы. Естественно считать, что количество транспортных единиц u_i , необходимое для перевозки продукта из i -го пункта, пропорционально объему перевозимого продукта. Простой

транспорта в i -м пункте производства измеряется величиной $a_i y_i = \alpha a_i x_i$, а суммарный простой транспорта равен $\alpha (a_1 x_1 + a_2 x_2 + \dots + a_n x_n)$.

Задача об организации снабжения формулируется теперь следующим образом. Необходимо обратить в минимум линейную форму L (1.5) при условиях (1.6)—(1.8) и дополнительном ограничении

$$\alpha (a_1 x_1 + a_2 x_2 + \dots + a_n x_n) \leq T.$$

Рассуждения, подобные предыдущим, уже не приводят к решению. В этом случае и тем более при необходимости учета новых ограничений планирование снабжения требует специальных методов, совокупность которых составляет содержание линейного программирования.

Можно, однако, указать еще один случай, когда независимо от количества условий решение задачи линейного программирования достигается элементарными средствами, не требующими трудоемких вычислений. Если область изменения переменных, определяемая условиями задачи, может быть изображена на плоскости (или в трехмерном пространстве), простые геометрические соображения быстро приводят к решению задачи.

Пусть, например, число пунктов производства в задаче о снабжении равно трем. Придадим для определенности параметрам задачи числовые значения

$$L = 3x_1 + 5x_2 + x_3,$$

$$x_1 + x_2 + x_3 = 10, \tag{a}$$

$$7x_1 + 4x_2 + 8x_3 \leq 76,$$

$$0 \leq x_1 \leq 6,$$

$$0 \leq x_2 \leq 3,$$

$$0 \leq x_3 \leq 8.$$

Введем новые переменные

$$y_1 = x_1,$$

$$y_2 = L = 3x_1 + 5x_2 + x_3. \tag{б}$$

Подставим в (а) и (б) y_1 вместо x_1 и, пользуясь этими соотношениями, выразим x_2 и x_3 через y_1 и y_2 . После этого перепишем задачу в новых переменных.

Требуется обратить в минимум линейную форму

$$L = y_2$$

при условиях

$$y_2 - y_1 \geq 14,$$

$$0 \leq y_1 \leq 6,$$

$$0 \leq y_2 - 2y_1 - 10 \leq 12,$$

$$0 \leq -y_2 - 2y_1 + 50 \leq 32.$$

На плоскости (y_1, y_2) область изменения переменных, ограниченная условиями задачи, представляет собой многоугольник. На рис. 1.1 эта область заштрихована. (Как видно из рисунка, условие $4x_3 = -y_2 - 2y_1 + 50 \geq 0$ оказалось излишним).

Самая нижняя точка области — вершина A — соответствует наименьшему значению y_2 и, следовательно, представляет собой минимум линейной формы L при заданных ограничениях.

Координаты точки A $y_1 = 1\frac{1}{3}$, $y_2 = L = 15\frac{1}{3}$.

Возвращаясь к исходным переменным, получаем решение задачи $x_1 = 1\frac{1}{3}$, $x_2 = \frac{2}{3}$, $x_3 = 8$, $L = 15\frac{1}{3}$.

Ясно, что такие же геометрические построения без труда приводят к решению задачи и при большем количестве ограничений. Все возникающие при этом случаи подробно рассмотрены в § 4.

Практические задачи, укладывающиеся в модель линейного программирования, как правило, содержат большое количество переменных и условий. Приведенные интуитивные и наглядные геометрические соображения в общем случае неэффективны. Задачи организации снабжения одного центра неоднородными продуктами или задачи снабжения нескольких центров могут быть быстро решены только при помощи специальных методов.

Рассмотрим, например, задачу об организации снабжения неоднородными продуктами, скажем, овощами или топливом. В этом случае могут быть указаны минимальные и предельные потребности в каждом отдельном

продукте, например, отдельно по картофелю и капусте (или углю и нефти) и условия взаимозаменяемости. Взаимозаменяемость должна быть охарактеризована коэффициентом, указывающим, сколько единиц одного продукта эквивалентно единице другого. Понятие эквивалентности в разных конкретных задачах может опреде-

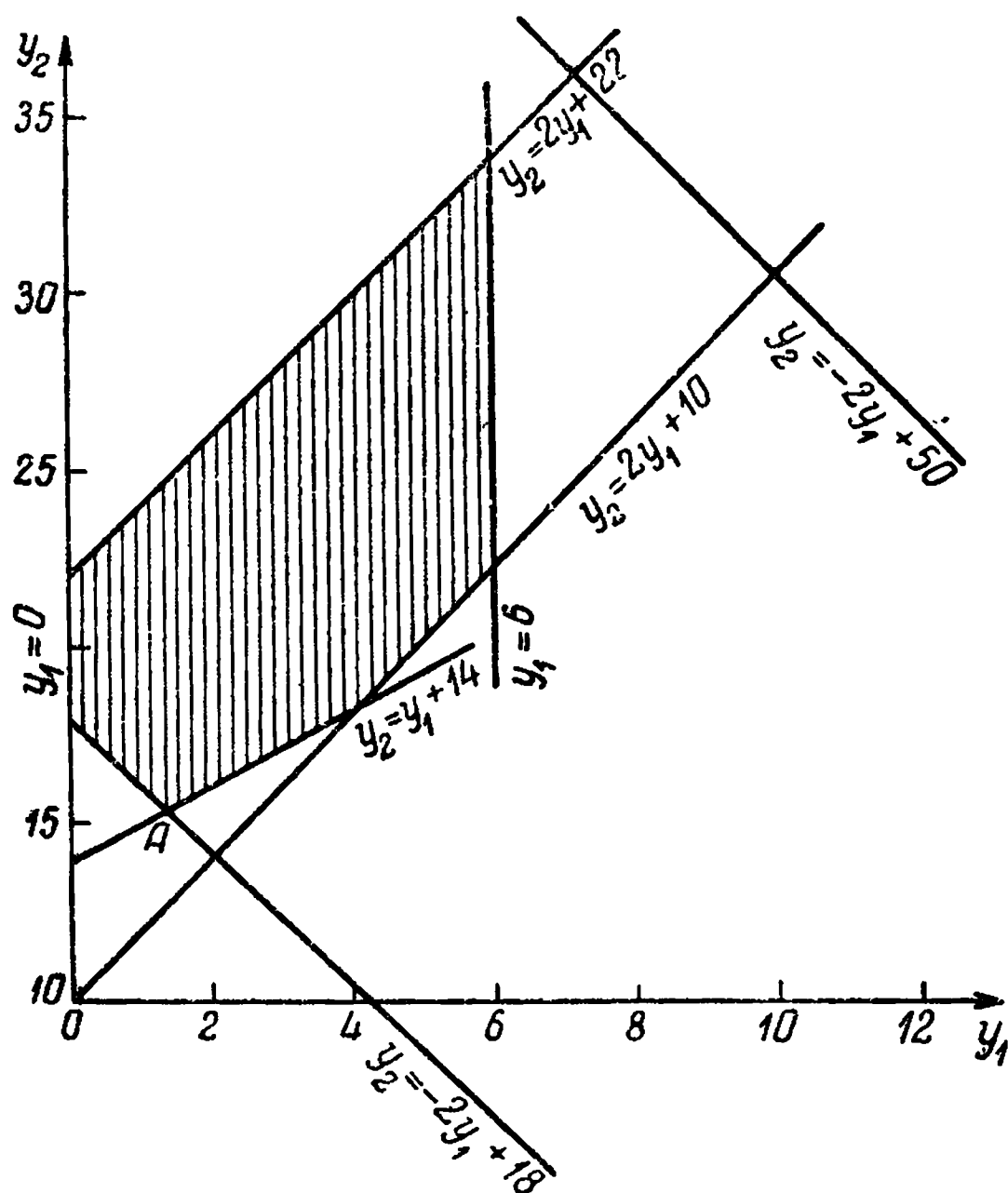


Рис. 1.1

ляться по-разному. В рассматриваемой задаче при оценке взаимозаменяемости продуктов можно исходить, например, из калорийности овощей (или топлива).

Сохраняя обозначения, принятые для случая однородного продукта, приведем математическую формулировку задачи.

Требуется обратить в минимум линейную форму

$$L = c_1^{(1)}x_1^{(1)} + \dots + c_n^{(1)}x_n^{(1)} + c_1^{(2)}x_1^{(2)} + \dots + c_n^{(2)}x_n^{(2)}$$

при условиях:

$$b_{\min}^{(1)} \leq x_1^{(1)} + \dots + x_n^{(1)} \leq b_{\max}^{(1)},$$

$$b_{\min}^{(2)} \leq x_1^{(2)} + \dots + x_n^{(2)} \leq b_{\max}^{(2)},$$

$$0 \leq x_i^{(1)} \leq \beta_i^{(1)},$$

$$0 \leq x_i^{(2)} \leq \beta_i^{(2)},$$

$$\mu_1 (x_1^{(1)} + \dots + x_n^{(1)}) + \mu_2 (x_1^{(2)} + \dots + x_n^{(2)}) = \mu.$$

Верхний индекс здесь означает номер продукта. Если некоторые из n пунктов являются пунктами производства только одного из рассматриваемых продуктов, то соответствующие значения $x_i^{(s)}$ заранее полагаются равными нулю. Последнее равенство в системе условий представляет собой условие взаимозаменяемости. Оно может быть истолковано, например, следующим образом: $\mu_1 (x_1^{(1)} + \dots + x_n^{(1)})$ — количество калорий, которое можно получить, сжигая $x_1 + \dots + x_n$ тонн угля, $\mu_2 (x_1^{(2)} + \dots + x_n^{(2)})$ — теплотворность доставленной нефти, а μ — требуемое количество теплоты. Конечно, не представляет никакого труда обобщение постановки задачи на случай снабжения нескольких центров произвольным количеством неоднородных продуктов.

2.2. Выбор системы вооружения. Приведенная ниже постановка задачи представляет собой некоторое видоизменение рассуждений, изложенных в статье Mag-schack'a и Miskey [58].

Для выполнения однотипных задач (например, для уничтожения однотипных целей) может быть использовано оружие различных классов. Требуется вычислить, в каком количестве следует производить комплексы оружия разных классов, чтобы обеспечить максимальную эффективность оружия при заданных ресурсах и ограниченных затратах. К такой постановке приводит, например, задача о распределении средств между заказами на производство тактической авиации разных типов, самолетов-снарядов и неуправляемых ракет оперативно-тактического назначения. Эту же задачу легко формулировать в терминах, связанных с выбором системы противотанкового вооружения или средств противовоздушной обороны.

Одно из основных допущений, принимаемых в настоящей постановке задачи, заключается в том, что

каждый комплекс действует только по своей группе целей.

Обозначим через x_j число комплексов оружия j -го типа, а через p_j — среднее число целей, поражаемых одним средством j -го типа. При принятом допущении показатель качества системы вооружения — математическое ожидание числа уничтоженных целей — может быть записан в виде

$$M = p_1 x_1 + p_2 x_2 + \dots + p_n x_n. \quad (1.9)$$

Обозначим стоимость одного комплекса j -го типа через r_j . Тогда ограничение количества комплексов, определяемое допустимыми расходами (b) на систему вооружения, может быть представлено следующим образом:

$$r_1x_1 + r_2x_2 + \dots + r_nx_n \leq b. \quad (1.10)$$

Комплекс вооружения — сложная система, производство которой ограничено не только возможными капиталовложениями, но и необходимостью использовать специальную аппаратуру и материалы, количество которых лимитировано.

Необходимость в тех или иных факторах (сырье, станочный парк, рабочая сила и т. д.) для комплексов различных типов различна. Пусть a_{ij} — количество единиц i -го фактора, необходимое для производства одного комплекса вооружения j -го типа, а b_i — ограничение по этому фактору. Тогда дополнительные условия, которым следует подчинить выбор x_j , записываются в виде

[illegible]

Здесь s — количество лимитирующих факторов.

Кроме перечисленных условий, ограничения по производству вооружения различного типа устанавливаются условиями хранения и эксплуатации, наличием или возможностью подготовки того или иного количества квалифицированного обслуживающего персонала и т. д. Таким образом, некоторые из искомых переменных x ,

(пусть это будут первые k из них) ограничены еще одной системой условий вида

$$x_j \leq d_j, j = 1, 2, \dots, k \leq n. \quad (1.12)$$

Ну и, конечно, все переменные (количество комплексов вооружения того или иного типа) должны принимать неотрицательные значения, т. е.

$$x_j \geq 0, j = 1, 2, \dots, n. \quad (1.13)$$

Вообще говоря, физический смысл искомых переменных требует выполнения еще одного условия — величины x_j должны быть целыми числами. Однако нарушение этого требования в задачах подобного рода при небольших значениях x_j не приводит, как правило, к существенным отклонениям от оптимума. При дробных x_j в качестве решения задачи следует принимать ближайšie к ним целые числа. В дальнейшем мы будем специально оговаривать те случаи, где требование целочисленности решения является существенным требованием.

Как видим, выбор рациональной системы вооружения сводится к обращению в максимум значения линейной формы M (1.9) при линейных условиях (1.10) — (1.13). Это задача линейного программирования.

Если выбор системы вооружения был бы ограничен только условиями (1.10), (1.12) и (1.13), то решение задачи можно было бы получить из тех же интуитивных соображений, что и в предыдущем пункте. Геометрические построения существенно упрощают решение задачи при любом количестве линейных ограничений вида (1.10) — (1.13), если речь идет об определении соотношения, в котором нужно производить оружие двух-трех типов. В общем же случае требуются более трудоемкие методы.

2.3. Размещение заказов и загрузка оборудования. Рассмотрим задачу о планировании производства сложного оборудования, состоящего из m элементов. Это может быть, например, система, состоящая из m приборов, или прибор из m деталей и т. д.

Заказы могут быть размещены на n различных предприятиях с различным станочным парком и разными возможностями. Требуется выяснить, какое распределение заказов между предприятиями обеспечит производ-

ство наибольшего количества комплектных систем в установленный срок. К аналогичным задачам приходим и тогда, когда речь идет о производстве одного прибора, изготовление деталей которого может быть распределено между различными станками предприятия или цеха. Рациональная загрузка оборудования должна обеспечить максимальный выпуск приборов.

Примем за единицу времени время, на которое рассчитана вся программа работы. Пусть в единицу времени на i -м предприятии (на i -м станке) можно изготовить a_{ij} элементов оборудования j -го сорта. Обозначим через x_{ij} время, в течение которого i -й исполнитель (предприятие или станок) загружен изготовлением элементов j -го сорта. Число j -х деталей, выпускаемых всеми исполнителями в единицу времени, равно

$$a_{1j}x_{1j} + a_{2j}x_{2j} + \dots + a_{nj}x_{nj}, \quad j = 1, 2, \dots, m. \quad (1.14)$$

Сдаче подлежат только готовые комплекты продукции. Количество комплектов оборудования, которое можно собрать из изготовленных деталей, лимитируется элементами того сорта, которых произведено меньше всего. Иными словами, число комплектов L , собранных в единицу времени, равно наименьшему из чисел (1.14), вычисленных для различных j :

$$L = \min_{1 \leq j \leq m} [a_{1j}x_{1j} + a_{2j}x_{2j} + \dots + a_{nj}x_{nj}].$$

Заказы будут размещены рационально и оборудование будет наилучшим образом загружено, если промежутки времени x_{ij} будут подобраны так, что ни одно предприятие (ни один станок) не будет простаивать, а величина L достигнет максимума. На формальном языке это значит, что необходимо обратить в максимум L при условиях:

$$x_{i1} + x_{i2} + \dots + x_{im} = 1, \quad i = 1, 2, \dots, n,$$

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m.$$

Первая группа условий требует, чтобы сумма времен, потраченных каждым исполнителем на изготовление элементов оборудования, равнялась единице, т. е. времени, на которое рассчитана вся программа работы. Это означает, что ни предприятия, ни станки не должны иметь холостого времени.

Вторая группа условий означает, что время загрузки любого исполнителя любым изделием не может быть отрицательным.

Задача на максимум минимума, к которой мы пришли, легко сводится к обычной задаче линейного программирования. Действительно, нетрудно заметить, что максимальное количество комплектов оборудования может быть произведено только в том случае, когда загрузка исполнителей обеспечивает производство одинакового количества деталей (элементов оборудования) всех сортов, т. е. решение задачи обязательно удовлетворяет условиям

$$\sum_{i=1}^n a_{i1}x_{i1} = \sum_{i=1}^n a_{i2}x_{i2} = \dots = \sum_{i=1}^n a_{im}x_{im}.$$

Действительно, если бы это было не так и хотя бы одна из этих сумм оказалась больше других, то за счет ее небольшого уменьшения можно было бы увеличить минимальную сумму и повысить таким образом количество комплектов. Но это означало бы, что выбрана не лучшая система размещения заказов и загрузки оборудования. Следовательно, изложенная задача может быть сформулирована еще и так.

Требуется обратить в максимум линейную форму

$$L = a_{11}x_{11} + a_{21}x_{21} + \dots + a_{n1}x_{n1}$$

при условиях:

$$\left. \begin{aligned} a_{12}x_{12} + a_{22}x_{22} + \dots + a_{n2}x_{n2} &= L, \\ a_{13}x_{13} + a_{23}x_{23} + \dots + a_{n3}x_{n3} &= L, \\ \dots & \\ a_{1m}x_{1m} + a_{2m}x_{2m} + \dots + a_{nm}x_{nm} &= L, \end{aligned} \right\}$$

$$\left. \begin{aligned} x_{11} + x_{12} + \dots + x_{1m} &= 1, \\ x_{21} + x_{22} + \dots + x_{2m} &= 1, \\ \dots & \\ x_{n1} + x_{n2} + \dots + x_{nm} &= 1, \end{aligned} \right\}$$

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, n, \quad j = 1, 2, \dots, m.$$

Мы пришли к задаче линейного программирования.

Как уже отмечалось, классическая математика не располагает методами, позволяющими решать подобные задачи при большом числе переменных и ограничений в приемлемые сроки. Задачи, аналогичные рассмотренным в этом параграфе, вызвали к жизни новое направление в математике — линейное программирование.

§ 3. КАНОНИЧЕСКАЯ ФОРМА ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

3.1. Все рассмотренные выше примеры укладываются в общий класс задач линейного программирования. Однако запись линейной формы и главным образом ограничений в разных задачах заметно различается. Как мы увидим дальше (гл. 2), можно указать много примеров, еще больше увеличивающих разнообразие форм записи задач линейного программирования. В одних случаях искомые переменные зависят от одного индекса, в других примерах — от двух. В одних задачах ограничения имеют вид равенств, в других случаях условия являются неравенствами. Ряд практических задач сводится также к смешанным условиям: часть ограничений — линейные уравнения, другие — линейные неравенства. Не во всех задачах требуется неотрицательность всех переменных. Такое разнообразие форм записи условий задач требует разработки специальных методов для решения отдельных классов задач и затрудняет исследование общих особенностей линейного программирования и создание общих методов и вычислительных алгоритмов. Поэтому естественно рассмотреть способ сведения любой задачи линейного программирования к наиболее простой и удобной для исследования форме.

Будем говорить, что задача линейного программирования записана в канонической форме, если она формулируется следующим образом.

Найти максимум (минимум) линейной формы

$$L = \bar{c}_1 x_1 + \dots + \bar{c}_n x_n$$

при условиях:

$$\bar{a}_{11}x_1 + \dots + \bar{a}_{1n}x_n = b_1,$$

• • • • •

$$\bar{a}_{m1}x_1 + \dots + \bar{a}_{m\bar{n}}x_{\bar{n}} = b_m,$$

$$x_1 \geq 0, \dots, x_{\bar{n}} \geq 0.$$

Задача линейного программирования, записанная в общем виде (1.1) — (1.4), может быть сведена к канонической форме.

Введем в задачу (1.1) — (1.4) дополнительные неотрицательные переменные $x_{n+1}, x_{n+2}, \dots, x_{n+(r-l)}$ [$x_i \geq 0$ для $i = n+1, \dots, n+(r-l)$]. Тогда ограничения (1.2), (1.3) эквивалентны следующим:

$$\begin{array}{rcl} a_{11}x_1 + \dots + a_{1n}x_n & & = b_1, \\ \dots & & \dots \\ a_{l1}x_1 + \dots + a_{ln}x_n & & = b_l, \\ a_{l+1,1}x_1 + \dots + a_{l+1,n}x_n + x_{n+1} & & = b_{l+1}, \\ a_{l+2,1}x_1 + \dots + a_{l+2,n}x_n & + x_{n+2} & = b_{l+2}, \\ \dots & & \dots \\ a_{r1}x_1 + \dots + a_{rn}x_n & + x_{n+(r-l)} & = b_r. \end{array}$$

В линейную форму дополнительные переменные входят с нулевыми коэффициентами. Если $t = n$, то общая задача линейного программирования уже сведена к каноническому виду. В этом случае $\bar{n} = n + (r - l)$; $m = r$. (Напомним, что t — число неотрицательных переменных в задаче (1.1) — (1.4).)

Пусть теперь смешанные условия сведены к системе k линейных равенств, но $t < n$, т. е. требование неотрицательности относится не ко всем искомым переменным.

Воспользуемся линейными уравнениями, ограничивающими выбор переменных, и выразим параметры, не связанные требованием неотрицательности, через все остальные переменные. Полученные выражения должны быть подставлены в линейную форму и в условия, не использованные при вычислениях.

В зависимости от вида линейной формы, характера линейных условий и соотношения между числом ограничений и количеством неотрицательных переменных вычисления приводят к одному из следующих случаев:

а) Задача линейного программирования неразрешима.

б) Общее количество переменных и условий сокращается. Задача линейного программирования принимает каноническую форму *).

Выкладки, обосновывающие приведенные выводы, громоздки. Мы их опустим. Для уяснения порядка вычислений при сведении задачи линейного программирования к канонической форме рассмотрим пример.

3.2. Пусть требуется найти максимум линейной формы

$$L = x_1 + x_2 + cx_3 + x_4$$

при условиях:

$$ax_1 + x_2 + x_3 - x_4 + 4x_5 = 4,$$

$$x_1 + ax_2 + x_3 - 2x_4 - 3x_5 = 3,$$

$$x_1 + x_2 - 2x_3 - 3x_4 + 2x_5 \leq 2,$$

$$x_1 + x_2 + ax_3 - 4x_4 - 4x_5 \leq 1,$$

$$x_4 \geq 0, x_5 \geq 0.$$

Для приведения условий задачи к однородной системе ограничений добавим две дополнительные неотрицательные переменные

$$x_6 \geq 0 \text{ и } x_7 \geq 0.$$

Система условий принимает вид

$$ax_1 + x_2 + x_3 - x_4 + 4x_5 = 4,$$

$$x_1 + ax_2 + x_3 - 2x_4 - 3x_5 = 3,$$

$$x_1 + x_2 - 2x_3 - 3x_4 + 2x_5 + x_6 = 2,$$

$$x_1 + x_2 + ax_3 - 4x_4 - 4x_5 + x_7 = 1,$$

$$x_4 \geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0.$$

Теперь следует переменные, не связанные условиями неотрицательности, выразить через остальные. Здесь возможны два случая. В одном случае можно выбрать три уравнения из четырех, которые могут быть разрешены

*) Задача линейного программирования, приведенная к канонической форме, не обязательно разрешима.

относительно x_1, x_2, x_3 . В другом случае этого сделать нельзя. Определитель, составленный из коэффициентов при x_1, x_2 и x_3 в I, II и IV уравнениях, равен $\Delta = (a - 1)^2(a + 2)$. При $a \neq 1$ и $a \neq -2$ можно выразить x_1, x_2 и x_3 через неотрицательные переменные. Примем $a = 0$. Вычислим x_1, x_2 и x_3 и подставим их выражения в линейную форму и третье условие, не использованное в расчетах.

Получим

$$x_1 = \frac{5}{2}x_4 + \frac{11}{2}x_5 - \frac{1}{2}x_7,$$

$$x_2 = 1 + \frac{3}{2}x_4 - \frac{3}{2}x_5 - \frac{1}{2}x_7,$$

$$x_3 = 3 - \frac{1}{2}x_4 - \frac{5}{2}x_5 + \frac{1}{2}x_7;$$

$$L = (1 + 3c) + \frac{1}{2}x_4(10 - c) + \\ + \frac{1}{2}x_5(8 - 5c) - \frac{1}{2}x_7(2 - c),$$

$$2x_4 + 11x_5 + x_6 - 2x_7 = 7,$$

$$x_4 \geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0.$$

Мы пришли к канонической форме задачи *).

Рассмотрим теперь другой случай, когда нельзя выбрать трех уравнений из четырех условий так, чтобы можно было, пользуясь ими, выразить не связанные требованием неотрицательности переменные через другие. Такой случай имеет место в рассматриваемом примере, если положить $a = -2$. Задача при этом формулируется следующим образом.

Требуется найти максимум линейной формы

$$L = x_1 + x_2 + cx_3 + x_4$$

при условиях:

$$-2x_1 + x_2 + x_3 - x_4 + 4x_5 = 4,$$

*) Постоянное слагаемое $1 + 3c$ в линейной форме можно отбросить, поскольку оно не влияет на выбор экстремума L .

$$\begin{aligned}x_1 - 2x_2 + x_3 - 2x_4 - 3x_5 &= 3, \\x_1 + x_2 - 2x_3 - 3x_4 + 2x_5 + x_6 &= 2, \\x_1 + x_2 - 2x_3 - 4x_4 - 4x_5 + x_7 &= 1, \\x_4 \geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0.\end{aligned}$$

Здесь нельзя выразить x_1 , x_2 и x_3 через неотрицательные переменные, но можно из первого и второго равенств выразить x_1 и x_2 через остальные переменные. Сделаем это и подставим полученные выражения в остальные условия задачи и в линейную форму. Получим

$$\left. \begin{aligned}L &= -7 + (2 + c)x_3 - 5x_4 + x_5, \\&\quad - 6x_4 + 3x_5 + x_6 = 9, \\&\quad - 7x_4 - 3x_5 + x_7 = 8, \\x_4 &\geq 0, x_5 \geq 0, x_6 \geq 0, x_7 \geq 0.\end{aligned} \right\} \quad (1.15)$$

Переменное x_3 в условиях отсутствует и, следовательно, не связано никакими ограничениями. Поэтому, если коэффициент при x_3 в линейной форме отличен от нуля, задача линейного программирования не будет иметь ограниченного решения. Если же $c = -2$, то система (1.15) представляет собой каноническую форму разрешимой задачи.

Указанный порядок вычислений не связан с особенностями приведенных примеров и может быть использован во всех случаях, когда требуется свести задачу линейного программирования к ее канонической форме.

§ 4. ГЕОМЕТРИЧЕСКИЙ СМЫСЛ ПРОСТЕЙШИХ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Рассмотрим задачи линейного программирования, допускающие геометрическое истолкование и решение на прямой, плоскости и в пространстве.

4.1. Начнем с задачи линейного программирования, линейная форма которой содержит только одно переменное

$$L = cx,$$

а условия имеют вид

$$\begin{aligned}ax &\leq b, \quad (a, b > 0), \\x &\geq 0.\end{aligned}$$

Как видно из рис. 1.2, линейная форма L достигает своих экстремальных значений на концах отрезка, на котором она согласно условиям задачи определена. Очевидно, что тот же вывод остается в силе, если несколько усложнить задачу и записать ее в следующем виде.

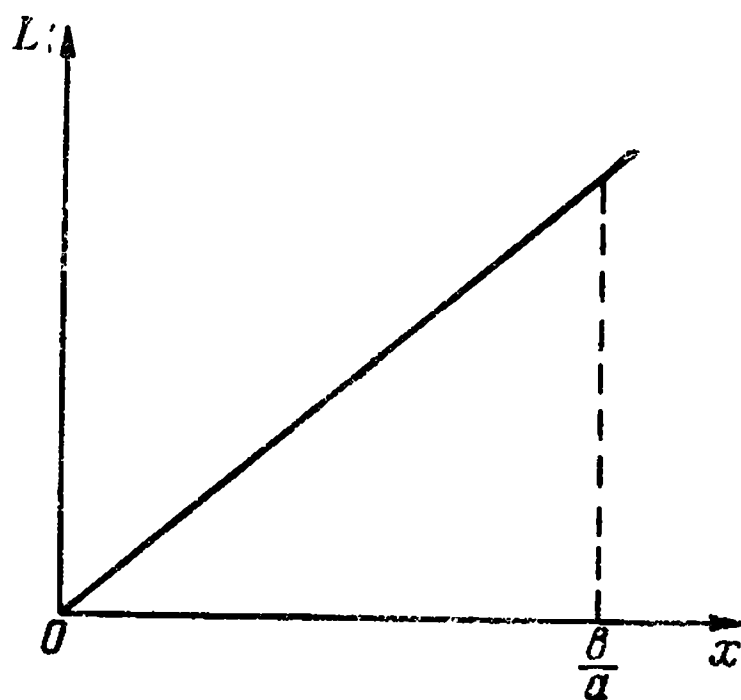


Рис. 1.2

Требуется обратить в максимум линейную форму,

$$L = c_1 x + c_2$$

при условиях

$$a_1 x \geq b_1,$$

$$a_2 x \leq b_2.$$

Вводя новую переменную $y = a_1 x - b_1$, приходим к прежней задаче.

Легко видеть, что все значения переменной x , для которых определена линейная форма L , т. е. все точки отрезка $\left[\frac{b_1}{a_1}, \frac{b_2}{a_2}\right]$ (и только они) выражаются через крайние точки отрезка по следующей формуле:

$$x = \lambda \frac{b_1}{a_1} + (1 - \lambda) \frac{b_2}{a_2}, \quad (1.16)$$

где

$$0 \leq \lambda \leq 1. \quad (1.17)$$

На соотношение (1.16) при условии (1.17) можно смотреть как на уравнение отрезка. Нетрудно сообразить, что ни одна из крайних точек отрезка не может быть выражена по формулам (1.16)—(1.17) через две другие точки отрезка. В дальнейшем мы увидим, что и в случае большего числа измерений аналоги крайних точек отрезка играют особую роль в задачах линейного программирования. Если $x = \lambda x_1 + (1 - \lambda) x_2$, где $0 \leq \lambda \leq 1$, то говорят, что точка x есть выпуклая линейная комбинация точек x_1 и x_2 . Таким образом, любая точка отрезка является выпуклой линейной комбинацией его концов.

4.2. Усложним несколько задачу и перейдем к случаю двух переменных. В этом случае линейная форма и условия задачи могут быть записаны в виде

$$L = c_1 x_1 + c_2 x_2, \quad (1.18)$$

$$\left. \begin{aligned} a_{11}x_1 + a_{12}x_2 &\leq b_1, \\ a_{21}x_1 + a_{22}x_2 &\leq b_2, \\ &\dots \dots \dots \\ a_{m1}x_1 + a_{m2}x_2 &\leq b_m, \end{aligned} \right\} \quad (1.19)$$

$$x_1 \geq 0, \quad x_2 \geq 0. \quad (1.20)$$

Ограничимся вначале двумя условиями ($m = 2$). Соотношение (1.18) представляет собой уравнение плоскости в осях x_1, x_2, L . Пусть на рис. 1.3 этому уравнению соответствует плоскость BOE . Она пересекает координатные плоскости LOx_1 и x_1Ox_2 соответственно по прямым BO и OE .

Соотношения

$$a_{11}x_1 + a_{12}x_2 = b_1,$$

$$a_{21}x_1 + a_{22}x_2 = b_2$$

представляют собой уравнения прямых KG и EF соответственно в плоскости x_1Ox_2 или уравнения вертикальных плоскостей BKG и AEF (плоскостей, параллельных оси OL) в трехмерном пространстве x_1x_2L .

Прямые EF и KG и оси координат Ox_1 и Ox_2 ограничивают область изменения значений x_1 и x_2 , из которых нужно будет выбрать координаты точки, обращающей в максимум (или минимум) линейную форму L . Пусть постоянные a_{ij} выбраны так, что область определения L представляет собой внутренность треугольника FDG . (В этом случае условие $x_1 \geq 0$ оказывается излишним.)

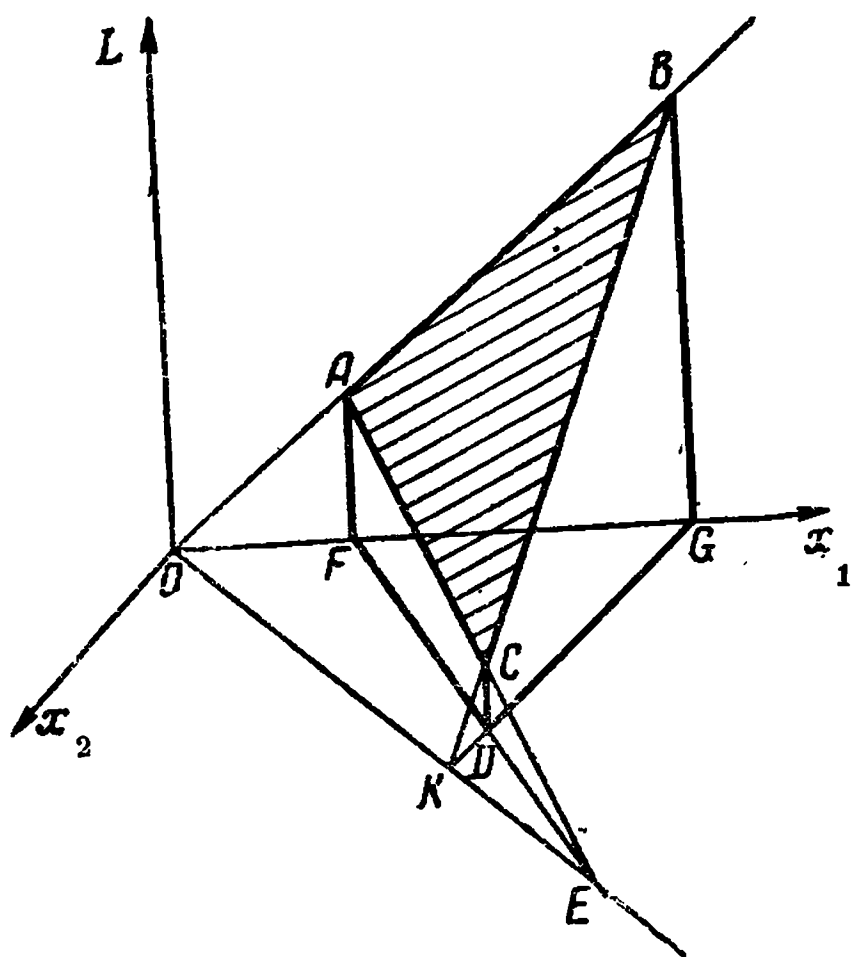


Рис. 1.3

Легко видеть, что здесь, как и в предыдущем случае, максимум (минимум) линейной формы L может быть достигнут в точках (x_1, x_2) , лежащих на границе области изменения переменных, т. е. в вершинах треугольника или на его сторонах. На рис. 1.3 $\max L$, равный длине отрезка BG , достигается в точке G ; $\min L$, равный длине CD , — в точке D . Если бы коэффициенты линей-

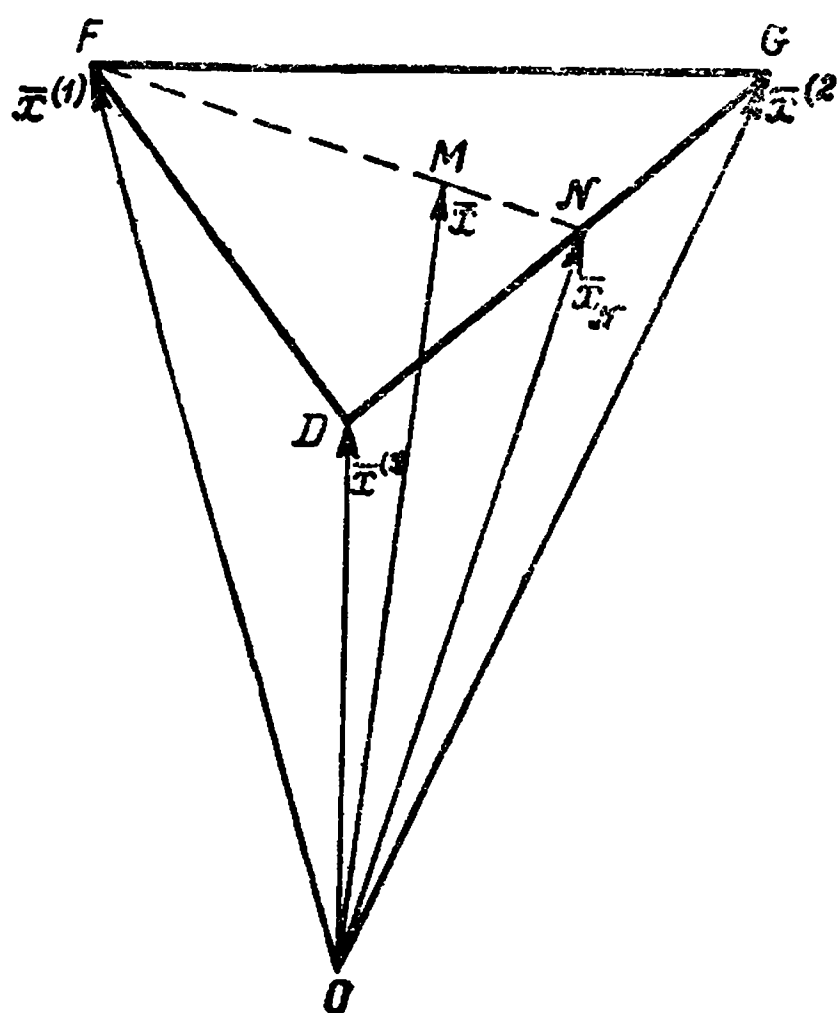


Рис. 1.4

ной формы и условия задачи оказались такими, что $AF = CD$, то $\min L$ достигался бы во всех точках, лежащих на отрезке FD .

Аналогию с одномерным случаем можно продолжить и далее. Обе прямоугольные координаты каждой точки области определения задачи линейного программирования (в данном случае $\triangle DFG$) могут быть выражены как одна и та же выпуклая линейная комбинация соответствующих координат край-

них точек — вершин треугольника. Это значит, что координаты x_i любой точки треугольника можно выразить через соответствующие координаты вершин $x_i^{(k)}$ по следующим формулам:

$$x_i = \lambda_1 x_i^{(1)} + \lambda_2 x_i^{(2)} + \lambda_3 x_i^{(3)}, \quad i = 1, 2,$$

где $\lambda_j \geq 0$, $j = 1, 2, 3$, $\lambda_1 + \lambda_2 + \lambda_3 = 1$.

В таких случаях говорят, что точка (x_1, x_2) является выпуклой линейной комбинацией точек $(x_1^{(k)}, x_2^{(k)})$.

Чтобы доказать приведенное утверждение, обратимся к рис. 1.4.

Пусть радиусы-векторы вершин $\triangle DFG$ соответственно $\bar{x}^{(1)}$, $\bar{x}^{(2)}$ и $\bar{x}^{(3)}$, $\bar{x}$ — радиус-вектор произвольной точки M треугольника. Точка M лежит на отрезке FN .

Поэтому в соответствии с предыдущими рассуждениями для одномерного случая имеет место соотношение

$$\bar{x} = \mu_1 \bar{x}^{(1)} + \mu_2 \bar{x}_N,$$

где $\bar{x}_N$ — радиус-вектор точки N , а

$$\mu_1 + \mu_2 = 1, \quad \mu_1 \geq 0, \quad \mu_2 \geq 0.$$

По той же причине

$$\bar{x}_N = \mu_3 \bar{x}^{(2)} + \mu_4 \bar{x}^{(3)},$$

$$\mu_3 + \mu_4 = 1, \quad \mu_3 \geq 0, \quad \mu_4 \geq 0.$$

Следовательно,

$$\bar{x} = \mu_1 \bar{x}^{(1)} + \mu_2 [\mu_3 \bar{x}^{(2)} + \mu_4 \bar{x}^{(3)}].$$

Вводя обозначения

$$\lambda_1 = \mu_1, \quad \lambda_2 = \mu_2 \mu_3, \quad \lambda_3 = \mu_2 \mu_4,$$

получаем

$$\bar{x} = \lambda_1 \bar{x}^{(1)} + \lambda_2 \bar{x}^{(2)} + \lambda_3 \bar{x}^{(3)}, \quad (1.21)$$

где $\lambda_1 \geq 0, \lambda_2 \geq 0, \lambda_3 \geq 0$,

$$\begin{aligned} \lambda_1 + \lambda_2 + \lambda_3 &= \mu_1 + \mu_2 \mu_3 + \mu_2 \mu_4 = \\ &= \mu_1 + (1 - \mu_1) \mu_3 + (1 - \mu_1) (1 - \mu_3) = 1. \end{aligned}$$

Соотношение (1.21), справедливое для векторов, имеет место и для их компонент.

Геометрически ясно, что радиус-вектор любой вершины треугольника не может быть выражен по формуле (1.21) через векторы других, отличных от нее точек треугольника. Таким образом, вершины треугольника играют ту же роль, что и крайние точки отрезка в одномерном случае.

Естественно, что приведенные рассуждения теряют смысл, если одна или несколько вершин уходят в бесконечность и многоугольник, на котором определена линейная форма, вырождается в неограниченную многоугольную область.

Если, рассматривая задачу линейного программирования для двух переменных, не ограничиваться случаем двух условий, а учесть все m условий, то после построений, аналогичных предыдущим, нетрудно получить следующий вывод.

Если множество значений (x_1, x_2) , удовлетворяющих условиям (1.19) и (1.20), ограничено, то оно представляет собой выпуклый многоугольник. Линейная форма достигает экстремума в вершине многоугольника. Если

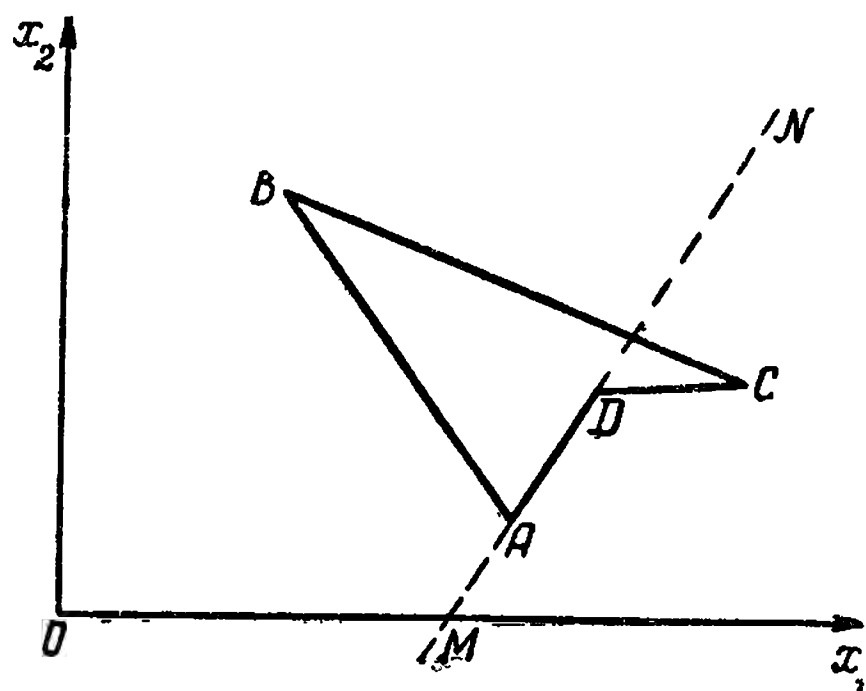


Рис. 1.5

максимум (или минимум) достигается одновременно в двух вершинах, то он достигается на всей стороне многоугольника, соединяющей эти вершины.

Напомним, что многоугольник (и вообще множество) называется выпуклым, если вместе с любыми двумя точками многоугольника (множеству)

принадлежит и весь отрезок, соединяющий эти точки.

Многоугольник, изображенный на рис. 1.5, не является выпуклым, так как отрезок, соединяющий точки A и C , не принадлежит многоугольнику. Ограничения типа (1.19) и (1.20) не могут привести к подобному многоугольнику.

Действительно, стороны многоугольника — это отрезки прямых, уравнения которых могут быть получены, если в условиях (1.19) и (1.20) заменить неравенства на равенства. Любое ограничение — неравенство означает, что область определения линейной формы рассматриваемой задачи лежит по одну сторону от каждой из прямых. Сторона AD многоугольника $ABCD$ (рис. 1.5) принадлежит прямой MN , по обе стороны от которой лежат внутренние точки многоугольника. Следовательно, многоугольник подобного типа не может служить областью определения линейной формы в задачах линейного программирования.

Рассуждая так же, как и выше в случае треугольника, легко показать, что все точки выпуклого k -угольника могут быть представлены как выпуклые линейные комбинации его вершин. Это значит, что радиус-вектор $\bar{x}$ любой точки многоугольника связан с радиусами-векторами его вершин соотношением вида

$$\bar{x} = \lambda_1 \bar{x}_1 + \lambda_2 \bar{x}_2 + \dots + \lambda_k \bar{x}_k,$$

где

$$\lambda_i \geq 0, \quad (i = 1, 2, \dots, k), \quad \lambda_1 + \lambda_2 + \dots + \lambda_k = 1.$$

Этими соотношениями можно воспользоваться для аналитического доказательства экстремального свойства вершин многоугольника в задачах линейного программирования.

4.3. Приведенные здесь геометрические рассуждения наглядны только при $n = 2$. Для того чтобы можно было обобщить полученные выводы, по крайней мере для $n = 3$, изменим несколько геометрическую интерпретацию задачи. Последующие построения позволят получить интуитивные основания для обобщения геометрически наглядных результатов на случай произвольного числа переменных.

Выводы, сформулированные выше для $n = 2$, можно, вообще говоря, получить, рассматривая плоскость $x_1 O x_2$ и не выходя в третье измерение.

Прямая KE на рис. 1.3 соответствует значению $L = 0$. Ее уравнение

$$c_1 x_1 + c_2 x_2 = 0.$$

Ясно, что уравнение семейства прямых, параллельных KE , можно записать в виде

$$c_1 x_1 + c_2 x_2 = L, \quad \text{где } L = \text{const.}$$

Параметр семейства пропорционален расстоянию от начала координат до соответствующей прямой семейства.

На рис. 1.6 изображен случай, когда область изменения линейной формы представляет собой многоугольник. Прямые, отсекающие многоугольник $OABCD$ на плоскости $x_1 O x_2$, соответствуют условиям (1.19) и (1.20), в которых неравенства заменены равенствами.

Штриховка указывает ту сторону прямой, по которую располагаются точки плоскости, удовлетворяющие неравенствам (1.19) и (1.20). Направление прямой MN определяется вектором (c_1, c_2) (вектор (c_1, c_2) перпендикулярен прямой MN). Коэффициенты c_1 и c_2 указывают также направление, в котором увеличивается линейная форма. Задача линейного программирования — вычисление координат точки, дающей экстремум линей-

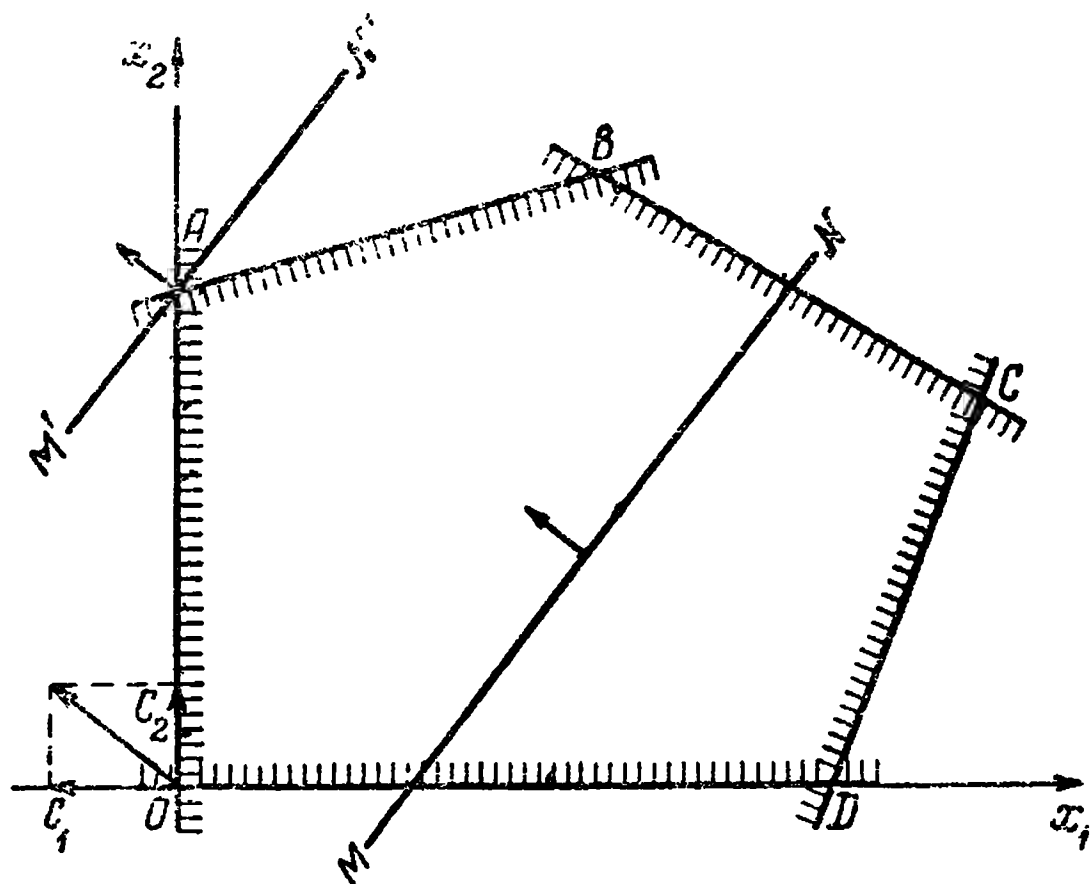


Рис. 1.6

ной форме (1.18) при условиях (1.19) и (1.20), — может быть геометрически (при $n = 2$) истолкована следующим образом.

Пересечем многоугольник условий — область определения линейной формы — прямой $L = c_1x_1 + c_2x_2$ и будем перемещать прямую параллельно самой себе в направлении увеличения L (если требуется вычислить максимум линейной формы) и в направлении уменьшения L (если требуется вычислить минимум линейной формы). При этом возможны два случая. В случае, изображенном на рис. 1.6, параллельный сдвиг приведет прямую в такое положение $M'N'$, когда у нее окажется только одна общая точка с многоугольником — вершина A . Эта точка определяет единственное решение задачи линейного программирования (на рис. 1.6 максимум линейной формы). Может оказаться (такой случай изображен на рис. 1.7), что прямая MN параллельна

одной или двум сторонам многоугольника. В таком случае экстремум достигается во всех точках соответствующей стороны многоугольника. На рис. 1.7 во всех точках стороны AB многоугольника $ABCDE$, параллельной прямой MN , достигается максимум, а во всех точках стороны $ED \parallel MN$ достигается минимум линейной формы. Таким образом, задача линейного программирования может иметь либо одно, либо бесконечное количество

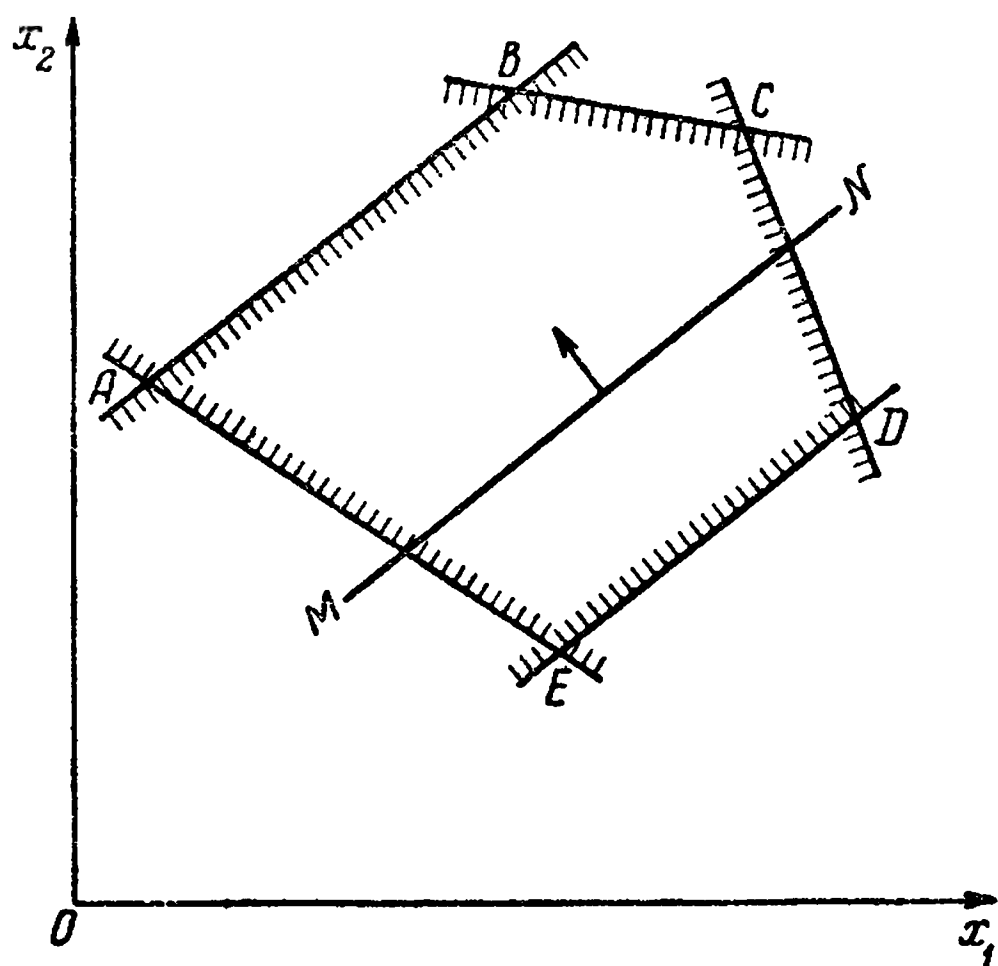


Рис. 1.7

решений. Если две вершины дают экстремум линейной формы, то и все точки отрезка, соединяющего эти вершины, определяют решение задачи линейного программирования.

Рис. 1.8 соответствует случаю, когда задача линейного программирования неразрешима, поскольку определяющие ее условия оказались противоречивыми.

На рис. 1.9 область определения линейной формы неограничена. В том случае, когда прямая $AB \parallel MN$, линейная форма достигает конечного экстремума во всех точках луча AB . Если изменять область определения линейной формы, поворачивая луч AB относительно точки A , то можно получить два случая—линейная форма может стать неограниченной при допустимых значениях переменных либо достигнуть максимума в единственной точке. Первый случай соответствует лучу AB' , изобра-

женному на рис. 1.9 пунктиром. Второму случаю соответствует штрих-пунктирный луч AB'' (рис. 1.9).

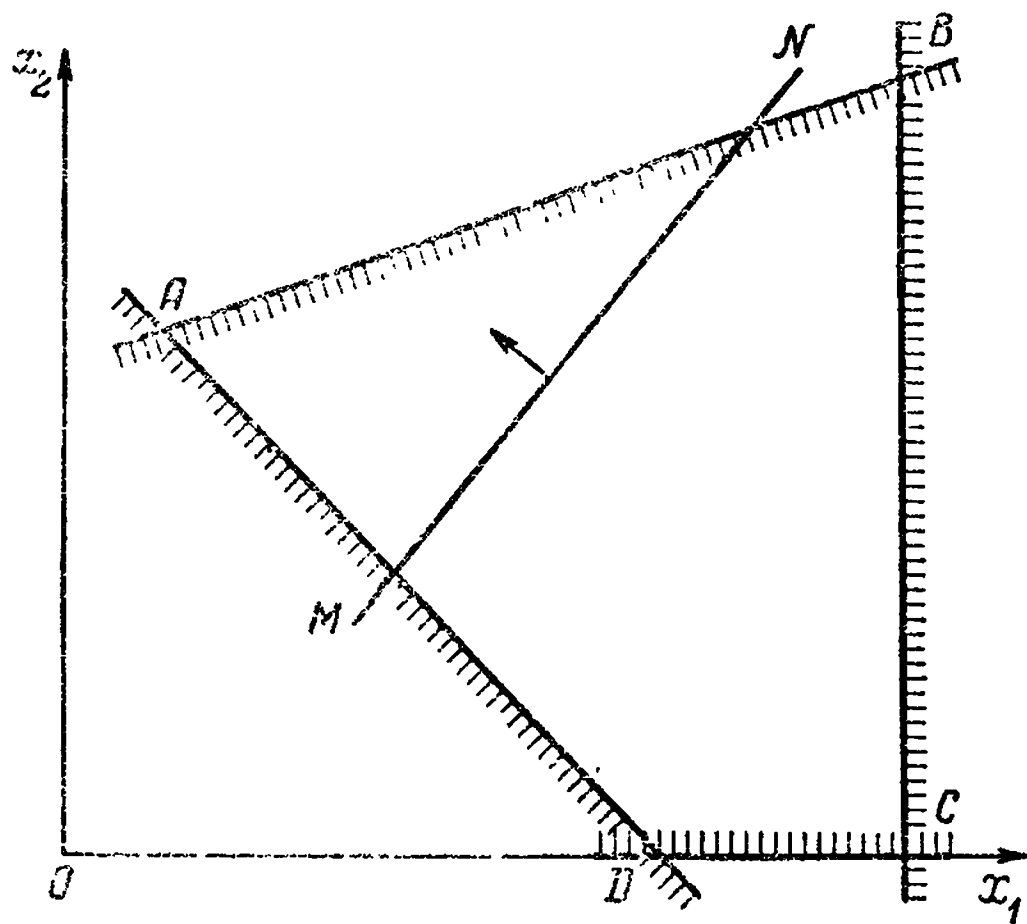


Рис. 1.8

4.4. До сих пор мы рассматривали геометрическое истолкование задачи линейного программирования с дву-

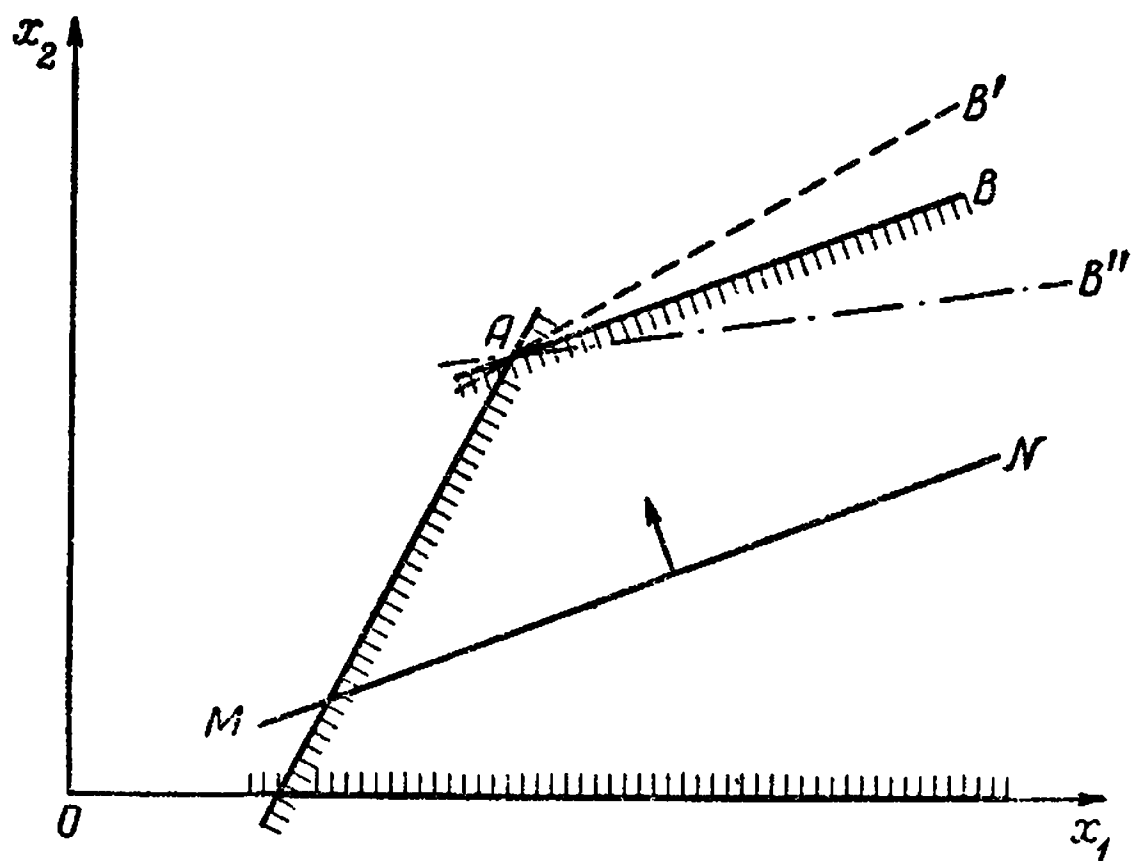


Рис. 1.9

мя переменными и m условиями — неравенствами вида (1.19). Если привести эту задачу к канонической форме, то число переменных новой задачи n станет равным

$m + 2$, а число условий (равенств) m . Таким образом, здесь $n - m = 2$. Легко видеть, что приведенное в предыдущем пункте геометрическое истолкование задачи линейного программирования остается справедливым и в том случае, когда n и m произвольны, но $n - m = 2$. Выразим все переменные через два из них, например через x_1 и x_2 , и перепишем линейную форму и условия задачи в следующем виде:

$$L = c_1'x_1 + c_2'x_2,$$

$$x_s = a_{1s}x_1 + a_{2s}x_2 - a_s, \quad s = 3, 4, \dots, n,$$

$$x_i \geq 0, \quad i = 1, 2, \dots, n.$$

Прямые $x_s = 0$ ($a_{1s}x_1 + a_{2s}x_2 = a_s$) и оси координат высекают на плоскости x_1x_2 выпуклый многоугольник. Условия $x_s \geq 0$ устанавливают направление штриховки сторон многоугольника. Так определяется область возможных значений пар чисел (x_1, x_2) , среди которых следует выбрать точки, обращающие линейную форму в максимум (или минимум). Коэффициенты линейной формы определяют семейство параллельных прямых и направление, в котором увеличивается L . Выберем из семейства любую прямую, пересекающую многоугольник, и будем смещать ее в сторону увеличения (уменьшения) линейной формы до тех пор, пока она еще будет содержать точки многоугольника. Предельное положение прямой определит максимальное (минимальное) значение линейной формы.

Такое же наглядное геометрическое истолкование задачи линейного программирования имеет место как для трех переменных, если условия имеют вид неравенств, так и в случае, когда число переменных превышает на 3 число условий ($n - m = 3$), а задача записана в канонической форме. Условия задачи высекают в пространстве выпуклый многогранник. Коэффициенты линейной формы определяют семейство параллельных плоскостей H и направление, в котором увеличивается L .

Для решения задачи линейного программирования следует перемещать плоскость H , пересекающую многогранник, в сторону увеличения линейной формы (если решается задача на максимум) или в сторону уменьшения L (если решается задача на минимум) до тех пор,

пока она еще содержит точки многогранника. Предельное положение плоскости определяет решение задачи.

Геометрические соображения подсказывают здесь, как и прежде, что экстремум достигается в крайних точках — в вершинах многогранника. Если экстремум достигается более чем в одной точке, то он достигается на всем ребре или на всей грани многогранника, параллельной плоскости, определяемой коэффициентами линейной формы задачи.

Любая точка выпуклого многогранника может быть представлена как выпуклая линейная комбинация его вершин, т. е. если $x_i^{(k)}$ — i -я координата k -й вершины, а x_i — i -я координата любой точки многогранника, то

$$x_i = \lambda_1 x_i^{(1)} + \lambda_2 x_i^{(2)} + \dots + \lambda_s x_i^{(s)}, \quad i = 1, 2, 3,$$

$$\lambda_k \geq 0, \quad \sum_{k=1}^s \lambda_k = 1.$$

Здесь s — число вершин многогранника.

Как и в предыдущих случаях, последние соотношения могут быть использованы для аналитического доказательства экстремальных свойств вершин многогранника. При большем числе переменных наглядность построений теряется и единственный путь установления основных результатов теории — аналитические рассуждения.

§ 5. ВЫПУКЛЫЕ МНОГОГРАННИКИ И ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ

5.1. Геометрическая интерпретация существенно облегчает исследование и решение задач линейного программирования. Отсюда естественно стремление к обобщению приведенного геометрического истолкования задачи на случай произвольного числа переменных. В этом и последующих параграфах нам понадобятся основные понятия n -мерного пространства. В частности, мы будем оперировать такими понятиями, как точка, вектор, линейная независимость векторов, отрезок, прямая, плоскость и т. п. Все они имеют наглядное геометрическое истолкование на плоскости и в трехмерном пространстве. Читатель при желании может ограничиться

двух- и трехмерным случаями. Строгое определение всех используемых понятий вынесено в § 1 гл. 6.

По-видимому, целесообразно принять в качестве определения многогранника в пространстве произвольного конечного числа измерений свойства многоугольников и многогранников, позволившие удобно и просто истолковать задачу линейного программирования на плоскости и в трехмерном пространстве. Можно ожидать, что при этом сохранится и описанный выше геометрический смысл задачи. Такие интуитивные соображения, конечно, не могут служить достаточной основой для последующих выводов. Ниже мы приведем необходимые формальные определения и доказательства.

Понятие выпуклой фигуры, выпуклого тела или выпуклого множества на плоскости и в пространстве — интуитивно ясное понятие. Тем не менее, следует привести формальное определение выпуклого множества, чтобы иметь возможность использовать его для геометрической интерпретации задачи линейного программирования в пространстве произвольного числа измерений.

Выпуклым множеством называется множество, которому вместе с любыми двумя его точками принадлежит отрезок, соединяющий эти точки. Напомним, что отрезок, соединяющий две точки $x^{(1)} = (x_1^{(1)}, \dots, x_n^{(1)})$ и $x^{(2)} = (x_1^{(2)}, \dots, x_n^{(2)})$, есть геометрическое место точек, координаты которых удовлетворяют уравнениям

$$\left. \begin{aligned} x_i &= \lambda x_i^{(1)} + (1 - \lambda) x_i^{(2)}, \\ i &= 1, 2, \dots, n, \quad 0 \leq \lambda \leq 1. \end{aligned} \right\} \quad (1.22)$$

Выпуклыми множествами являются треугольник, круг, параллелепипед, шар и т. д.

Крайними точками выпуклого множества называются точки, которые не лежат внутри отрезков, соединяющих какие бы то ни было точки множества. Другими словами, крайними точками называются точки, координаты которых не могут быть выражены через соответствующие координаты любых других двух точек множества по формулам (1.22).

Выпуклое множество может иметь конечное или бесконечное число крайних точек. Некоторые множества не

содержат их вовсе. Так, например, для шара все точки сферы — крайние точки, а для квадрата или параллелепипеда крайними точками являются только их вершины. Прямая и плоскость не имеют ни одной крайней точки. Наличие конечного числа крайних точек является характеристическим свойством многоугольника на плоскости и многогранника в пространстве. Как следствие отсюда можно было бы получить, что все точки выпуклых многогранников и многоугольников ($n = 2, 3$) являются выпуклыми линейными комбинациями их вершин. Это свойство многогранника естественно принять в качестве определения выпуклого многогранника в пространстве произвольного конечного числа измерений.

Будем называть выпуклым многогранником замкнутое*) ограниченное выпуклое множество, если совокупность его крайних точек конечна.

Крайние точки выпуклого многогранника будем в дальнейшем называть его вершинами.

Можно доказать (это будет сделано в § 1 гл. 6), что любая точка выпуклого многогранника, определенного, как указано выше, может быть представлена в виде выпуклой линейной комбинации его вершин. Поэтому иногда говорят, что многогранник порождается своими вершинами.

5.2. Ниже будет обобщен на случай любого числа переменных вывод о том, что область определения задачи линейного программирования (если она ограничена) является выпуклым многогранником. Кроме того, в этом пункте приведены необходимые и достаточные условия, которым удовлетворяют координаты вершин многогранника, определяемого системой ограничений задачи линейного программирования.

Напомним, что задача линейного программирования состоит в вычислении экстремума линейной формы (1.1) при условиях (1.2) — (1.4).

Условия (1.2) — (1.4) ограничивают область определения линейной формы (1.1).

Теорема 1. *Множество точек (векторов) $X = (x_1, x_2, \dots, x_n)$, удовлетворяющих условиям (1.2) — (1.4), является выпуклым.*

*) Определение понятия замкнутое множество см. в § 1 гл. 6.

Доказательство. Пусть $X' = (x'_1, x'_2, \dots, x'_n)$ и $X'' = (x''_1, x''_2, \dots, x''_n)$ две произвольные точки области определения задачи (1.1) — (1.4). Покажем, что при любом λ из отрезка $[0, 1]$ точка

$$Y = (y_1, y_2, \dots, y_n) = \{\lambda x'_1 + (1 - \lambda) x''_1, \lambda x'_2 + (1 - \lambda) x''_2, \dots, \lambda x'_n + (1 - \lambda) x''_n\}$$

также принадлежит области определения рассматриваемой задачи. Для этого достаточно установить, что координаты точки Y удовлетворяют системе ограничений (1.2) — (1.4).

Рассмотрим произвольное условие системы (1.2) — (1.4)

$$a_{i1}x_1 + \dots + a_{in}x_n \leq b_i.$$

По предположению

$$a_{i1}x'_1 + \dots + a_{in}x'_n \leq b_i,$$

$$a_{i1}x''_1 + \dots + a_{in}x''_n \leq b_i.$$

Умножая первое из этих неравенств на $\lambda \geq 0$, второе — на $1 - \lambda \geq 0$ и складывая результаты, получаем

$$a_{i1}[\lambda x'_1 + (1 - \lambda) x''_1] + \dots + a_{in}[\lambda x'_n + (1 - \lambda) x''_n] \leq \lambda b_i + (1 - \lambda) b_i = b_i,$$

т. е.

$$a_{i1}y_1 + \dots + a_{in}y_n \leq b_i.$$

Точно так же доказывается, что координаты точки Y удовлетворяют всем остальным равенствам и неравенствам системы (1.2) — (1.4). Таким образом, вместе с любыми двумя точками X' и X'' области определения задачи (1.1) — (1.4) принадлежит произвольная точка Y вида $Y = \lambda X' + (1 - \lambda) X''$, где $0 \leq \lambda \leq 1$. В соответствии с определением это означает, что область определения задачи линейного программирования является выпуклым множеством.

В дальнейшем рассмотрении крайние точки области определения задачи играют особую роль. В связи с этим целесообразно иметь критерий, позволяющий проверить, является ли данная точка крайней точкой множества,

определяемого условиями (1.2)—(1.4). В двух- и трехмерном случаях такие критерии могут быть получены из геометрических соображений. Для пространств большего числа измерений следует перевести эти соображения на аналитический язык.

Ниже устанавливаются необходимые и достаточные условия, которым удовлетворяют крайние точки области определения задачи линейного программирования.

Теорема 2. *Для того чтобы точка $X = (x_1, \dots, x_n)$ из области определения задачи (1.1)—(1.4) была крайней, необходимо и достаточно, чтобы X удовлетворяла не менее, чем n независимым линейным ограничениям системы (1.2)—(1.4), как равенствам.*

Систему линейных ограничений

$$\sum_{j=1}^n d_{ij}x_j = g_i, \quad i=1, 2, \dots, l;$$

$$\sum_{j=1}^n d_{ij}x_j \leq g_i, \quad i=l+1, \dots, r$$

будем называть независимой, если векторы $d_i = (d_{i1}, \dots, d_{in})$ ($i=1, 2, \dots, r$) линейно независимы. (Определение линейной независимости векторов дано в § 1 гл. 6.)

Любая точка области определения линейной формы удовлетворяет l равенствам (1.2) (они предполагаются линейно независимыми) и $(r-l)+t$ неравенствам (1.3) и (1.4). Для того чтобы точка X была крайней точкой рассматриваемого множества, требуется согласно сформулированной теореме, чтобы ее координаты обратили в равенства по меньшей мере $n-l$ независимых условий из $(r-l)+t$ неравенств (1.3) и (1.4). При этом все n равенств, которым удовлетворяет X , должны быть линейно независимы.

Линейное уравнение определяет в n -мерном пространстве гиперплоскость^{*)}. Поэтому геометрический смысл рассматриваемого характеристического свойства крайних точек множества (1.2)—(1.4) заключается в том, что каждая из них является пересечением не менее чем n независимых гиперплоскостей.

^{*)} Гиперплоскость в n -мерном пространстве — аналог плоскости в трехмерном пространстве. Строгое определение гиперплоскости приведено в § 1 гл. 6.

Доказательство теоремы 2. 1. Докажем достаточность утверждения. Пусть сформулированная теорема неверна: точка X является решением системы n линейно независимых уравнений, но не является крайней точкой множества (1.2) — (1.4). Тогда X может быть представлено в виде

$$X = \lambda X^{(1)} + (1 - \lambda) X^{(2)},$$

где $0 < \lambda < 1$, а $X^{(1)} \neq X^{(2)}$ принадлежат области определения линейной формы задачи и не совпадают с X . Покажем, что $X^{(1)}$ и $X^{(2)}$ должны удовлетворять тем же n линейно независимым уравнениям, что и точка X . Допустим, что точка $X = (x_1, \dots, x_n)$ удовлетворяет одному из условий системы (1.2) — (1.4) как равенству. Пусть, например,

$$a_{i1}x_1 + \dots + a_{in}x_n = b_i.$$

По условию

$$a_{i1}x_1' + \dots + a_{in}x_n' \leq b_i,$$

$$a_{i1}x_1'' + \dots + a_{in}x_n'' \leq b_i.$$

Имеем

$$\begin{aligned} a_{i1}x_1 + \dots + a_{in}x_n &= a_{i1}[\lambda x_1' + (1 - \lambda)x_1''] + \\ &+ \dots + a_{in}[\lambda x_n' + (1 - \lambda)x_n''] = \lambda(a_{i1}x_1' + \dots + a_{in}x_n') + \\ &+ (1 - \lambda)(a_{i1}x_1'' + \dots + a_{in}x_n'') = b_i. \end{aligned}$$

Учитывая, что $\lambda > 0$, получаем из последнего равенства

$$\begin{aligned} a_{i1}x_1' + \dots + a_{in}x_n' &= \\ &= \frac{1}{\lambda} [b_i - (1 - \lambda)(a_{i1}x_1'' + \dots + \\ &+ a_{in}x_n'')] \geq \frac{1}{\lambda} [b_i - (1 - \lambda)b_i] = b_i. \end{aligned}$$

Полученное неравенство не противоречит исходному только в том случае, если

$$a_{i1}x_1' + \dots + a_{in}x_n' = b_i.$$

Аналогично проверяется, что

$$a_{i1}x_1'' + \dots + a_{in}x_n'' = b_i.$$

Таким образом, наборы чисел $X = (x_1, \dots, x_n)$, $X^{(1)} = (x_1^{(1)}, \dots, x_n^{(1)})$ и $X^{(2)} = (x_1^{(2)}, \dots, x_n^{(2)})$ удовлетворяют одной и той же системе из n независимых линейных уравнений. Поэтому $X = X^{(1)} = X^{(2)}$, что противоречит допущению. Следовательно, точка X является крайней точкой множества. Достаточность условий теоремы доказана.

2. Докажем необходимость условий теоремы. Пусть X — крайняя точка области определения линейной формы и пусть компоненты X обращают в равенства $k < n$ соотношений из системы (1.2) — (1.4). Все остальные ограничения при этом имеют вид строгих неравенств. В системе уравнений, которым удовлетворяет X , уравнений меньше, чем неизвестных. Поэтому соответствующая система однородных уравнений имеет по крайней мере одно нетривиальное решение $\tilde{X} = (\tilde{x}_1, \dots, \tilde{x}_n)$ ($\tilde{X} \neq 0$). Точки $X^{(1)} = X + \varepsilon \tilde{X}$ и $X^{(2)} = X - \varepsilon \tilde{X}$ удовлетворяют той же системе неоднородных уравнений, что и X . Остальным условиям системы (1.3) — (1.4) точка X удовлетворяет как строгим неравенствам. Поэтому при достаточно малом ε точки $X^{(1)}$ и $X^{(2)}$ удовлетворяют тем же неравенствам. Таким образом, $X^{(1)}$ и $X^{(2)}$ удовлетворяют всем ограничениям (1.2) — (1.4). При этом

$$X = \frac{X^{(1)} + X^{(2)}}{2}, \text{ а } X^{(1)} \neq X^{(2)}, \text{ так как } \tilde{X} \neq 0.$$

Но это противоречит допущению о том, что X — крайняя точка области определения линейной формы (1.1). К такому же выводу приходим и в тех случаях, когда X удовлетворяет n или более уравнениям, но среди них нет n линейно независимых. Исключая из рассмотрения уравнения, оказывающиеся следствиями других равенств, приходим к случаю $k < n$. Необходимость условий теоремы доказана.

Область определения задачи линейного программирования задается N условиями (если задача записывается в форме (1.1) — (1.4), $N = r + t$).

Согласно теореме 2 каждой крайней точке области определения задачи соответствует n линейно независимых условий (из общего числа N ограничений задачи), которым координаты крайней точки удовлетворяют как

равенствам. Различным крайним точкам соответствуют различные системы линейно независимых соотношений. Это следует из того, что система из n линейно независимых уравнений с n неизвестными всегда имеет единственное решение.

При общем количестве условий, равном N , количество различных систем из n линейно независимых равенств не превышает C_N^n . Следовательно, область определения задачи (1.1) — (1.4) не может содержать более C_N^n крайних точек.

Итак, доказано, что область определения произвольной задачи линейного программирования выпукла, а совокупность ее крайних точек конечна. В тех случаях, когда эта область ограничена, она представляет собой по определению выпуклый многогранник *).

Будем в дальнейшем называть область определения задачи линейного программирования **многогранником условий**.

Напомним, что крайние точки многогранника мы называли его вершинами.

Следует отметить, что термин «многогранник условий» не всегда соответствует геометрическому смыслу области определения задачи. Если условия задачи противоречивы, многогранник условий не содержит ни одной точки. Если область определения линейной формы оказывается неограниченной (некоторые вершины уходят в бесконечность), многогранник условий вырождается в неограниченное многогранное тело. В практических задачах такие случаи встречаются редко. Поэтому термин «многогранник условий», естественный для наиболее важного и часто встречающегося случая, целесообразно сохранить как название области определения линейной формы задачи.

5.3. В двух- и трехмерном случаях геометрическая наглядность подсказывала, что экстремум линейной формы достигается в вершинах многоугольника или многогранника — области определения линейной формы задачи. Если экстремум достигается более чем в одной вершине, то он достигается на ребре или грани многогран-

*) На доказательстве замкнутости области определения задачи мы здесь не останавливаемся. Читатель, знакомый с понятием замкнутого множества, сумеет провести это доказательство самостоятельно.

ника. Этот результат может быть обобщен на случай произвольного числа переменных.

Теорема 3. *Линейная форма (1.1) достигает экстремума в вершинах многогранника условий. Если экстремум достигается не в одной, а в нескольких вершинах, то он достигается и на всем многограннике, порожденном этими вершинами.*

Доказательство. Будем предполагать область определения линейной формы ограниченной. Пусть для определенности рассматриваемая задача состоит в определении максимума линейной формы

$$L = L(X) = L(x_1, x_2, \dots, x_n).$$

Линейная форма, как и всякая непрерывная функция, определенная на ограниченном замкнутом множестве, достигает максимума на многограннике условий. Допустим, что максимум линейной формы достигается в точке X . Точка X , как любая точка многогранника, может быть представлена в виде выпуклой линейной комбинации его вершин (как уже отмечалось, это утверждение доказывается в § 1 гл. 6)

$$X = \lambda_1 X^{(1)} + \dots + \lambda_s X^{(s)},$$

$$\lambda_i \geq 0, \quad \lambda_1 + \dots + \lambda_s = 1.$$

В силу линейности формы (1.1) значение L в точке X , в которой достигается экстремум, связано со значениями формы в вершинах $X^{(i)}$ таким же соотношением, каким связаны координаты этих точек, т. е.

$$L(X) = \lambda_1 L(X^{(1)}) + \dots + \lambda_s L(X^{(s)}),$$

$$\lambda_i \geq 0, \quad \sum_{i=1}^s \lambda_i = 1.$$

Перепишем последнее условие, сохранив лишь слагаемые с $\lambda_i > 0$. Число таких слагаемых $s_1 \leq s$. Перенумеровав соответствующим образом вершины, запишем

$$L(X) = \sum_{i=1}^{s_1} \lambda_i L(X^{(i)}), \quad \lambda_i > 0, \quad \sum_{i=1}^{s_1} \lambda_i = 1.$$

Отсюда при любом k ($1 \leq k \leq s_1$)

$$L(X^{(k)}) = \frac{1}{\lambda_k} L(X) - \frac{1}{\lambda_k} \sum_{\substack{i=1 \\ i \neq k}}^{s_1} \lambda_i L(X^{(i)}).$$

Максимум линейной формы достигается по условию в точке X . Поэтому

$$\begin{aligned} L(X^{(k)}) &\geq \frac{1}{\lambda_k} L(X) - \frac{1}{\lambda_k} \sum_{\substack{i=1 \\ i \neq k}}^{s_1} \lambda_i L(X) = \\ &= \frac{1}{\lambda_k} \left[1 - \sum_{\substack{i=1 \\ i \neq k}}^{s_1} \lambda_i \right] L(X) = L(X). \end{aligned}$$

Последнее соотношение имеет смысл наряду с условием $L(X) \geq L(X^{(k)})$ (означающим, что максимум L достигается в точке X) только в том случае, когда $L(X^{(k)}) = L(X)$. Следовательно, если экстремум линейной формы достигается в какой-то точке многогранника условий, то он обязательно достигается и в некоторых его вершинах.

Если экстремум достигается в r вершинах, то он достигается и на всем многограннике, порожденном этими вершинами. Действительно, пусть $\max L = L(X^{(1)}) = \dots = L(X^{(r)})$. Тогда

$$\begin{aligned} L(X) &= L(\mu_1 X^{(1)} + \dots + \mu_r X^{(r)}) = \\ &= \mu_1 L(X^{(1)}) + \dots + \mu_r L(X^{(r)}) = \\ &= (\mu_1 + \dots + \mu_r) \max L = \max L. \end{aligned}$$

Теорема доказана.

Полученный результат сформулирован и доказан для ограниченной области изменения переменных задачи. Мы видели, однако, что система соотношений (1.2) — (1.4) может определять и неограниченную область, которая тем не менее содержит только конечное число крайних точек. В § 8 гл. 3 будет показано, что если задача линейного программирования разрешима, то и в этом случае найдется крайняя точка области определения

задачи, в которой линейная форма достигает экстремального значения.

Доказательство второго утверждения теоремы 3 для неограниченной области определения переменных задачи ничем не отличается от приведенного выше.

Заметим, что случай неограниченной области изменения переменных задачи линейного программирования, как правило, не представляет практического интереса. Этот случай можно всегда устранить, «отсекая» область определения линейной формы некоторой достаточно удаленной гиперплоскостью так, чтобы все исходные крайние точки остались внутри полученной ограниченной области. Для задачи линейного программирования, все переменные которой предполагаются неотрицательными, ограничение области определения линейной формы может быть достигнуто, например, добавлением условия

$$x_1 + x_2 + \dots + x_n \leq b_{m+1},$$

где b_{m+1} превышает сумму координат любой из исходных крайних точек. Поскольку экстремум разрешимой задачи линейного программирования достигается в крайних точках, то указанное ограничение области определения задачи не повлияет на экстремальное значение линейной формы. Как мы увидим в дальнейшем, основные методы линейного программирования позволяют решать задачу и в тех случаях, когда область определения линейной формы неограничена.

Итак, разрешимая задача линейного программирования имеет либо единственное решение, либо бесконечное множество решений. В последнем случае вся совокупность решений определяется конечным числом точек. Таким образом, решение общей задачи линейного программирования сводится к перебору вершин выпуклого многогранника, определяемого условиями задачи.

§ 6. ВЕКТОРЫ УСЛОВИЙ И ВЕКТОР ОГРАНИЧЕНИЙ

6.1. В § 3 было показано, что каждая задача линейного программирования может быть записана в канонической форме. При такой записи от всех переменных требуется неотрицательность и все условия вида (1.2), (1.3) превращаются в равенства.

В этом параграфе будем иметь дело с задачами линейного программирования, записанными в канонической форме,

$$L = c_1 x_1 + c_2 x_2 + \dots + c_n x_n, \quad (1.1)$$

[illegible]

$$x_j \geq 0, \quad j=1, 2, \dots, n. \quad (1.24)$$

Введем понятие вектора условий. Под j -м вектором условий A_j будем понимать вектор, компоненты которого являются коэффициентами при x_j в условиях (1.23). Через B обозначим вектор, компоненты которого представляют собой ограничения b_i в правых частях условий задачи линейного программирования. Вектор B будем называть вектором ограничений.

$$A_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{mj} \end{pmatrix}; \quad B = \begin{pmatrix} b_1 \\ b_2 \\ \vdots \\ b_m \end{pmatrix}; \quad j = 1, 2, \dots, n.$$

Система условий (1.23) может быть записана в виде единого условия в векторной форме

$$A_1x_1 + A_2x_2 + \dots + A_nx_n = B. \quad (1.25)$$

На это равенство можно смотреть, как на разложение вектора ограничений B по векторам условий A_j (x_j являются коэффициентами этого разложения).

В дальнейшем нам понадобится следующее важное утверждение.

Теорема 4. Для того чтобы совокупность неотрицательных чисел $x_1, x_2, \dots, x_n$, удовлетворяющая системе (1.23), определяла координаты вершины многогранника условий, необходимо и достаточно, чтобы система векторов A_j при ненулевых x_j была линейно независима.

Доказательство. 1. Мы уже видели (теорема 2), что необходимое и достаточное условие того, что совокупность чисел $(x_1, \dots, x_n)$ определяет вершину многогранника условий, состоит в том, что эти числа дол-

жны обращать в равенства не менее n линейно независимых условий задачи линейного программирования.

Пусть среди m условий (1.23) имеется лишь l линейно независимых. Тогда точка $X = (x_1, \dots, x_n)$ может быть вершиной многогранника условий в том и только в том случае, если $(n - l)$ её координат равны нулю.

Изменим нумерацию переменных и условий так, чтобы l первых равенств, определяющих вершину многогранника, имели вид

$$\sum_{j=1}^l a_{ij} x_j = b_i; \quad i = 1, 2, \dots, l, \quad (1.26)$$

а остальные $(n - l)$ фиксировали бы нулевые координаты

$$x_{l+1} = x_{l+2} = \dots = x_n = 0. \quad (1.27)$$

Линейная независимость условий (1.26) и (1.27) означает, что определитель, составленный из коэффициентов при x_j , отличен от нуля, т. е.

$$\begin{vmatrix} a_{11} & \dots & a_{1l} & 0 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ a_{l1} & \dots & a_{ll} & 0 & \dots & 0 \\ 0 & \dots & 0 & 1 & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & \dots & 0 & 0 & \dots & 1 \end{vmatrix} \neq 0.$$

Это соотношение возможно лишь, если

$$\begin{vmatrix} a_{11} & \dots & \dots & a_{1l} \\ a_{21} & \dots & \dots & a_{2l} \\ \cdot & \cdot & \cdot & \cdot \\ a_{l1} & \dots & \dots & a_{ll} \end{vmatrix} \neq 0.$$

Отсюда следует, что система векторов

$$\tilde{A}_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{lj} \end{pmatrix}, \quad j = 1, 2, \dots, l$$

линейно независима. Но тогда линейно независима и любая система из l векторов, первые l компонент которых равны соответствующим компонентам $\tilde{A}_j$.

В частности, векторы

$$A_j = \begin{pmatrix} a_{1j} \\ a_{2j} \\ \vdots \\ a_{lj} \\ \vdots \\ a_{mj} \end{pmatrix}, \quad j = 1, 2, \dots, l$$

удовлетворяют равенству

$$A_1 x_1 + \dots + A_l x_l = B$$

и линейно независимы. Итак, если точка X является крайней точкой области определения задачи линейного программирования, то система векторов условий при ненулевых координатах X линейно независима. Необходимость утверждения доказана.

2. Пусть теперь, наоборот, система линейно независимых векторов $A_1, \dots, A_l$ удовлетворяет условию

$$A_1 x_1 + \dots + A_l x_l = B$$

при положительных значениях $x_1, \dots, x_l$. Это означает, что из матрицы

$$A = (A_1, \dots, A_l) = \begin{pmatrix} a_{11} & \dots & a_{1l} \\ \vdots & & \vdots \\ a_{m1} & \dots & a_{ml} \end{pmatrix}$$

можно выделить квадратную подматрицу $l \times l$, определитель которой отличен от нуля. Пусть эта подматрица содержит верхние l строк матрицы A .

Следовательно, координаты точки X удовлетворяют l независимым равенствам вида

$$\sum_{j=1}^l a_{ij} x_j = b_i, \quad i = 1, 2, \dots, l.$$

Кроме того, $n - l$ координат точки X равны нулю

$$x_{l+1} = \dots = x_n = 0.$$

Нетрудно убедиться в том, что определитель полученной системы равенств отличен от нуля (см. первую часть доказательства теоремы). Следовательно, эти равенства линейно независимы. Обращаясь к теореме 3, получаем, что точка $(x_1, x_2, \dots, x_l, 0, 0, \dots, 0)$ является вершиной многогранника условий задачи. Достаточность условий теоремы доказана.

6.2. Объединяя результаты настоящего и предыдущего параграфов, приходим к следующему утверждению.

Пусть точка $X = (x_1, \dots, x_n)$ удовлетворяет всем условиям задачи линейного программирования. Тогда следующие три предложения равносильны.

1. Точка X является вершиной многогранника условий.

2. Координаты точки X удовлетворяют n линейно независимым равенствам из системы (1.23)—(1.24).

3. Векторы условий при ненулевых координатах X линейно независимы.

Установленное соответствие вершин многогранника условий и систем линейно независимых векторов условий является исходным пунктом для построения алгоритмов решения задач линейного программирования.

Мы видели, что линейная форма достигает экстремума в вершинах многогранника условий. Перебор вершин многогранника условий эквивалентен, таким образом, перебору линейно независимых систем векторов A_j . Различные методы линейного программирования по-разному упорядочивают этот перебор.

§ 7. ГЕОМЕТРИЧЕСКИЕ ИНТЕРПРЕТАЦИИ ОБЩЕЙ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

7.1. Как мы уже видели, геометрические соображения оказываются весьма полезными для исследования структуры задач линейного программирования.

В этом параграфе рассматриваются две различные геометрические интерпретации общей задачи линейного программирования. Первая из них применительно к простейшим задачам линейного программирования приводилась в § 4.

Здесь мы распространим эту интерпретацию на случай произвольного числа условий и переменных.

вий M расположен по одну сторону от полученной гиперплоскости и имеет с ней общие точки, каждая из которых придает линейной форме (1.1) экстремальное (максимальное или минимальное в зависимости от постановки задачи) значение. Если задача имеет единственное решение, то многогранник M и предельная гиперплоскость обладают лишь одной общей точкой, совпадающей с некоторой вершиной многогранника.

В общем случае совокупность решений задачи, являясь пересечением предельной гиперплоскости и многогранника условий, представляет собой выпуклый многогранник. Таким образом, полученная предельная гиперплоскость определяет всю совокупность решений задачи линейного программирования.

В некоторых задачах множество M может оказаться неограниченным. Поскольку это множество высекается с помощью конечного числа гиперплоскостей, оно представляет собой неограниченную многогранную область с конечным числом крайних точек (вершин). В данном случае могут представиться две возможности:

1) Гиперплоскость, определяемая линейной формой (1.1), будучи передвинута в сторону возрастания (убывания) L как угодно далеко, пересекает многогранное тело M .

2) Существует предельное положение гиперплоскости, пройдя через которое, она уже не будет иметь общих точек с многогранным телом M .

В первом случае задача линейного программирования неразрешима, так как ее линейная форма неограничена на множестве M (сверху для задачи на максимум, снизу для задачи на минимум).

Во втором случае задача имеет решение, определяемое предельным положением рассматриваемой гиперплоскости. Здесь, как и прежде, полная совокупность решений задачи совпадает с общей частью многогранного тела M и предельной гиперплоскости. Иногда некорректная постановка задачи приводит к противоречивости ее условий. Геометрически этот случай соответствует тому, что область определения задачи M вырождается в пустое множество, т. е. в множество, не содержащее ни одной точки.

7.2. Приведем еще одну геометрическую интерпретацию задачи, весьма удобную для пояснения особенно-

стей основных методов линейного программирования.

Рассмотрим задачу линейного программирования в канонической форме (1.1), (1.23), (1.24).

Введем новые переменные $y_1, y_2, \dots, y_m, y_{m+1}$:

$$\left. \begin{aligned} y_1 &= a_{11}x_1 + \dots + a_{1n}x_n, \\ &\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ &\cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \quad \cdot \\ y_m &= a_{m1}x_1 + \dots + a_{mn}x_n, \\ y_{m+1} &= c_1x_1 + \dots + c_nx_n. \end{aligned} \right\} \quad (1.28)$$

Соотношения (1.28) определяют преобразование n -мерного пространства переменных $x_1, x_2, \dots, x_n$ в $(m+1)$ -мерное пространство переменных $y_1, y_2, \dots, y_{m+1}$. При этом преобразовании положительные полуоси $Ox_1, \dots, Ox_n$ n -мерного пространства переходят в лучи, исходящие из начала координат. Действительно, координаты положительной полуоси Ox_1 удовлетворяют условиям

$$x_2 = x_3 = \dots = x_n = 0.$$

Поэтому образ Ox_1 в пространстве $y_1, \dots, y_{m+1}$ определяется соотношениями

$$y_1 = a_{11}x_1, \quad y_2 = a_{21}x_1, \dots, y_m = a_{m1}x_1, \quad y_{m+1} = c_1x_1, \\ 0 \leq x_1 < \infty$$

и, следовательно, совпадает с некоторым лучом $(m+1)$ -мерного пространства. Аналогично положительная полуось Ox_k

$$x_1 = \dots = x_{k-1} = x_{k+1} = \dots = x_n = 0, \quad 0 \leq x_k < \infty,$$

переходит в луч

$$y_1 = a_{1k}x_k, \dots, y_m = a_{mk}x_k, \quad y_{m+1} = c_k x_k, \quad 0 \leq x_k < \infty.$$

На рис. 1.10 изображены в осях y_1, y_2, y_3 (применительно к случаю $m=2$) лучи Λ_i , соответствующие положительным полуосям координат в пространстве $x_1, \dots, x_n$. Совокупность лучей Λ_i определяет выпуклый много-

гранный конус — образ «октанта» $x_i \geq 0$ ($i=1, 2, \dots, n$) пространства x -ов. «Октант» $x_i \geq 0$ — выпуклое множество. Нетрудно показать, что линейное преобразование переводит выпуклое множество в выпуклое. Отсюда вытекает выпуклость многогранного конуса. Полученный для $m=2$ вывод легко обобщается на случай произвольного m .

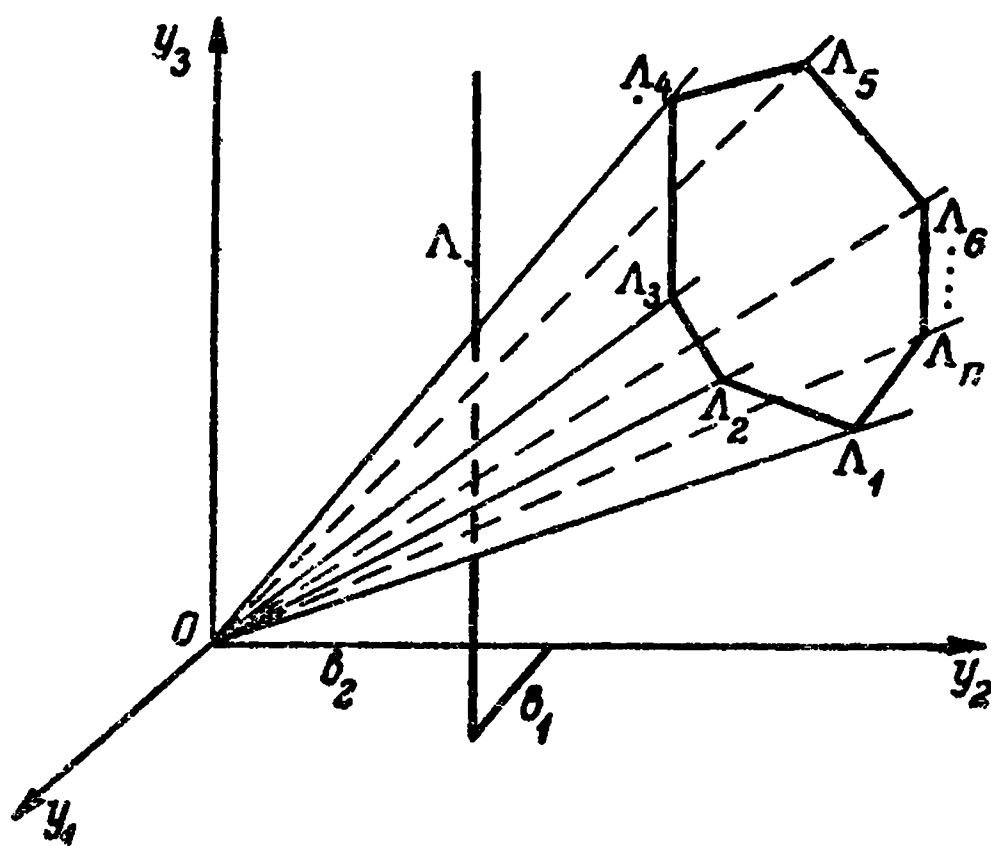


Рис. 1.10

Для геометрической интерпретации задачи линейного программирования в $(m+1)$ -мерном пространстве при $m>2$ необходимо ввести понятие выпуклого многогранного конуса, пригодное для пространств произвольного числа измерений. Допустим, что имеется некоторая система векторов (точек) $R_1, R_2, \dots, R_s$. Выпуклым многогранным конусом, порожденным системой $R_1, \dots, R_s$, назовем множество точек P , для которых возможно представление

$$P = \mu_1 R_1 + \mu_2 R_2 + \dots + \mu_s R_s,$$

где $\mu_i \geq 0$, $i=1, 2, \dots, s$.

Заметим, что в трехмерном случае (рис. 1.10) характеристическое свойство, принятое нами для аналитического определения выпуклого многогранного конуса, легко устанавливается из элементарных геометрических соображений.

Обозначим через $\bar{A}_k$ $(m + 1)$ -мерный вектор с компонентами $a_{1k}, a_{2k}, \dots, a_{mk}, c_k$. Таким образом, первые m компонент вектора

$$\bar{A}_k = \begin{pmatrix} a_{1k} \\ a_{2k} \\ \vdots \\ a_{mk} \\ c_k \end{pmatrix}$$

равны соответствующим составляющим вектора условий A_k , а последняя совпадает с коэффициентом линейной формы c_k . Будем в дальнейшем называть вектор $\bar{A}_k$ расширенным вектором условий. Расширенные векторы условий определяют направления лучей Λ_i — образов положительных полуосей пространства переменных $x_1, x_2, \dots, x_n$.

Из соотношений (1.28) следует, что положительный «октант» пространства x -ов переходит в выпуклый многогранный конус $(m + 1)$ -мерного пространства, порожденный расширенными векторами условий

$$\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n.$$

7.3. Для исследования особенностей задачи линейного программирования представляет интерес не образ положительного «октанта», а образ той его части, которая является областью определения рассматриваемой задачи. Поэтому целесообразно наряду с лучами Λ_i (образами положительных полуосей) рассмотреть еще прямую Λ , определяемую соотношениями

$$\begin{aligned} y_1 &= b_1, \quad y_2 = b_2, \dots, \quad y_m = b_m, \\ y_{m+1} &= \lambda \quad (-\infty < \lambda < \infty), \end{aligned}$$

где b_k — составляющие вектора ограничений B .

Прямая Λ параллельна оси Oy_{m+1} , а ее проекцией на подпространство $\{y_1, \dots, y_m\}$ является вектор B . На рис. 1.10 прямая Λ вертикальна. Ее проекция на плоскость y_1Oy_2 имеет компоненты (b_1, b_2) .

Покажем, что в пространстве $y_1, \dots, y_{m+1}$ образом области определения задачи линейного программирования является отрезок — общая часть прямой Λ и вы-

выпуклого многогранного конуса. В самом деле, для любой точки $Y = (y_1, y_2, \dots, y_m, y_{m+1})$ отрезка справедливы равенства

$$y_1 = a_{11}x_1 + \dots + a_{1n}x_n = b_1,$$

$$y_m = a_{m1}x_1 + \dots + a_{mn}x_n = b_m,$$

поскольку отрезок принадлежит прямой Λ . Кроме того, для всех точек отрезка

$$x_1 \geq 0, \quad x_2 \geq 0, \dots, \quad x_n \geq 0,$$

поскольку они принадлежат многогранному конусу — образу положительного «октанта» пространства x -ов. Таким образом, каждой точке рассматриваемого отрезка соответствует некоторая точка области определения линейной формы задачи. Очевидно, что справедливо и обратное: каждая точка области определения задачи переходит в точку отрезка, являющегося пересечением многогранного конуса и прямой Λ .

Каждой точке отрезка соответствует некоторое значение $y_{m+1} = \lambda$. Но

$$y_{m+1} = c_1x_1 + \dots + c_nx_n$$

определяет значение линейной формы L . Следовательно, параметр λ отрезка соответствует величине линейной формы. Если прямая Λ содержит внутренние точки выпуклого многогранного конуса, то, как нетрудно проверить, граница конуса имеет с этой прямой не более двух точек пересечения. Одна из них соответствует максимальному значению линейной формы, другая — минимальному. Если прямая Λ касается конуса, т. е. принадлежит одной из его граней, то «верхняя» точка касания определяет максимум линейной формы, а «нижняя» — минимум.

Таким образом, задача линейного программирования — задача определения экстремума линейной формы (1.1) при условиях (1.23) и (1.24) — сводится к определению крайней («верхней» в случае максимума и «нижней» в случае минимума) точки пересечения прямой Λ с выпуклым многогранным конусом.

Если прямая Λ не пересекает выпуклый многогранный конус, порожденный расширенными векторами усло-

вий $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n$, то область определения задачи линейного программирования не содержит ни одной точки. В этом случае задача, естественно, неразрешима. Если общая часть прямой Λ и выпуклого многогранного конуса неограничена («сверху» для задачи на максимум, «снизу» для задачи на минимум), то рассматриваемая задача неразрешима вследствие неограниченности линейной формы в области ее определения.

§ 8. ЭКОНОМИЧЕСКАЯ ИНТЕРПРЕТАЦИЯ И ТЕРМИНОЛОГИЯ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

8.1. Основную массу приложений линейного программирования составляют экономические задачи. Собственно линейное программирование и возникло из потребностей экономики. Поэтому целесообразно наряду с геометрическим истолкованием задачи линейного программирования ввести также ее экономическую интерпретацию.

Рассмотрим вопросы планирования работы предприятия или группы предприятий, производящих некоторый однородный продукт. Производство продукта требует ряда категорий сырья, определенного станочного парка, рабочей силы различной квалификации, энергии, топлива, транспорта и т. д. Будем называть перечисленные факторы производственными. Пусть число факторов, определяющих технологию производства, равно m . Как правило, каждый из них имеется на предприятиях в ограниченном количестве.

Каждый реализуемый набор производственных факторов обеспечивает определенный выход продукции в единицу времени.

Будем называть способом производства систему $m + 1$ чисел $[(m + 1)$ -мерный вектор], определяющих затраты каждого производственного фактора в единицу времени и выпуск продукции за то же время.

В каждой конкретной задаче можно указать много способов производства, различным образом использующих производственные факторы для производства продукции. Можно себе мыслить следующий порядок составления плана работы предприятия.

Заранее отрабатывается (или предполагается заданным) ряд способов производства — ряд наборов затрат каждого из производственных факторов в единицу вре-

мени и обеспечиваемый ими выпуск продукции. В зависимости от того, сколько времени предприятие работает по тому или иному из исходных способов производства, будет затрачиваться различное количество производственных факторов разных категорий и производиться различное количество продукции. Задача планирования производства состоит, таким образом, в определении продолжительности работы по отдельным способам производства.

Наилучшим планом нужно будет признать тот план, при котором затраты по каждому производственному фактору не превысят допустимых, а продукция предприятия достигнет наибольшего возможного значения. Естественно, чем больше проанализировано и отработано исходных способов производства, тем лучшим будет оптимальный план.

Формализация задачи планирования предприятия в указанном выше смысле требует следующих вполне естественных в широком круге вопросов предположений.

Если I набор затрат производственных факторов обеспечивает в единицу времени c_1 единиц продукции, а II — c_2 , то работа предприятия в течение x_1 единиц времени в соответствии с I набором затрат и x_2 — по второму набору обеспечит $c_1x_1 + c_2x_2$ единиц продукции. Конечно, нет никакой необходимости предполагать, что промежутки времени x_1 и x_2 следуют один за другим. Предприятие может одновременно использовать различные способы производства, заменяя в определенные моменты времени отдельные способы производства на другие. Промежутки времени x_1 и x_2 , таким образом, играют здесь роль удельных весов, с которыми исходные способы производства участвуют в выбранном для предприятия способе производства. Время перехода с одного способа производства на другой — время холостого хода — в рассматриваемой постановке задачи не учитывается.

8.2. Переведем задачу о планировании работы предприятия на математический язык.

Обозначим через c_j количество единиц продукции, выпущенной предприятием при работе в течение единицы времени по j -му способу производства, когда затраты 1-го производственного фактора составляют a_{1j} , 2-го a_{2j} , ..., i -го a_{ij} , ..., m -го a_{mj} . Пусть, кроме того, числа

раньше были названы соответственно векторами условий и вектором ограничений. В экономической интерпретации компоненты вектора условий определяют затраты отдельных производственных факторов в единицу времени для соответствующего исходного способа производства, а компоненты вектора ограничений — запасы отдельных факторов, ограничивающие их расход.

В терминах производственной задачи векторы A_j ($j=1, 2, \dots, n$) и вектор B естественней называть соответственно векторами затрат и вектором запасов.

j -й способ производства определяется вектором $\bar{A}_j$, первые m компонент которого совпадают с составляющими вектора условий A_j , а последняя составляющая равна c_j — соответствующему объему продукции в единицу времени. В соответствии с введенным в § 7 определением будем называть $\bar{A}_j$ расширением вектора A_j . Расширением вектора B является вектор $\bar{B}$, первые m компонент которого совпадают с составляющими вектора ограничений, а $(m+1)$ -я компонента определяет экстремальное значение линейной формы L . Расширенные векторы полностью характеризуют способы производства, определяя как затраты, так и продукцию.

Заметим, что расширенные векторы условий $\bar{A}_j$, соответствующие отдельным способам производства, определяют направления ребер выпуклого многогранного конуса, введенного в предыдущем параграфе для геометрической интерпретации задачи линейного программирования.

Набор чисел $x_1, \dots, x_n$, удовлетворяющих ограничениям, будем в дальнейшем называть планом, независимо от конкретного содержания задачи.

Систему чисел $x_1, \dots, x_n$, удовлетворяющих условиям задачи и обращающих ее линейную форму в максимум (или при соответствующей постановке задачи — в минимум), будем называть оптимальным планом или решением.

При ограниченной области определения задачи каждый план соответствует некоторой точке многогранника условий. Вершины многогранника условий играют особую роль при определении экстремума линейной формы. С другой стороны, вершины многогранника полностью

его определяют. Поэтому естественно планы, отвечающие вершинам многогранника условий, называть опорными планами^{*)}. Векторы затрат, отвечающие положительным компонентам опорных планов, образуют систему линейно независимых векторов. То же относится к соответствующим расширенным векторам, определяющим исходные способы производства.

Систему линейно независимых векторов затрат (векторов условий), отвечающих положительным компонентам опорного плана, будем называть базисом плана или просто базисом. В дальнейшем (§ 7 гл. 3) понятие базиса будет несколько уточнено.

Линейная форма разрешимой задачи достигает экстремума в вершинах многогранника условий. В терминах производственной задачи это значит, что для каждой разрешимой задачи линейного программирования существует опорный план, являющийся решением этой задачи.

§ 9. ОБЩАЯ ХАРАКТЕРИСТИКА МЕТОДОВ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

9.1. Мы уже видели, что классические методы дифференциального исчисления не могут быть использованы для определения экстремума линейной формы при линейных условиях. Доказано также, что решение задачи линейного программирования сводится в общем случае к перебору конечного количества точек n -мерного пространства, где n — число переменных. Однако, как легко видеть, даже в относительно простых задачах число операций, с которыми связано вычисление координат вершин многогранника условий и сравнение значений линейной формы в них, выражается астрономическими цифрами. Действительно, в задаче линейного программирования с n переменными и m условиями необходимо проверить линейную независимость C_n^m систем линейных уравнений. Каждая из систем содержит m уравнений с m неизвестными. Те из систем, которые окажутся линейно независимыми, должны быть решены. Из всех наборов решений следует выделить наборы с неотрицательными

^{*)} Отметим, что термин «опорный план» не связан с вводимым в гл. 6 понятием «опорная гиперплоскость».

компонентами. Так будут найдены координаты вершин многогранника условий. Далее следует вычислить значения линейной формы во всех вершинах многогранника и сравнить их между собой.

Как видим, перебор вершин содержит существенно больше чем C_n^m операций, причем отнюдь не элементарных (при $n = 50$, $m = 25$ $C_n^m > 10^{14}$). В одной из классических задач линейного программирования — в проблеме выбора (она будет рассмотрена в гл. 5) — легко можно указать все вершины соответствующего многогранника условий. Суть проблемы выбора в следующем. Задана квадратная таблица с n строками и n столбцами. Требуется выбрать по одному элементу в каждой строке и каждом столбце так, чтобы сумма их оказалась максимальной. Эта задача, имеющая самые разнообразные приложения, оказывается задачей линейного программирования. Количество вершин соответствующего многогранника условий равно $n!$ Таким образом, непосредственное решение проблемы выбора связано со сравнением $n!$ величин. Для вычисления каждой из этих величин необходимо произвести n сложений. При $n > 15—20$ такое количество операций нельзя провести за обозримый срок на самых быстродействующих современных машинах.

По формуле Стирлинга

$$n! \approx \left(\frac{n}{e}\right)^n \sqrt{2\pi n}.$$

При $n = 20$ $n! > 2 \cdot 10^{18}$.

Машине, выполняющей 1 миллион операций в секунду (таких машин пока еще нет), потребуется более 50 000 лет для того, чтобы перебрать вершины многогранника, определяемого этой относительно простой задачей.

При $n = 30$ количество опорных планов проблемы выбора превысит 10^{31} . По-видимому, легче пересчитать все песчинки на земном шаре, чем перебрать вершины многогранника условий для этой задачи.

Практика требует решения задач типа проблемы выбора для значений n , значительно превышающих 20 и 30. Приведенные примеры дают необходимое представление о роли упорядочивания перебора вершин в задачах линейного программирования.

При упорядоченном переборе вершин для решения задачи линейного программирования не требуется сравнения значений линейной формы во всех вершинах многогранника условий. Методы линейного программирования существенно сокращают также число сравнений, необходимых для того, чтобы установить, достигается ли на данной вершине экстремум линейной формы.

Заметим, что методы линейного программирования обеспечивают решение проблемы выбора при числе операций порядка n^3 . Существующие уже вычислительные машины потребуют считанное количество секунд для решения проблемы выбора при $n = 20$ и 30 и даже при $n = 100$.

9.2. За последние годы многие авторы в разных странах разработали большое количество методов линейного программирования.

Все методы линейного программирования можно разбить на две группы. Методы первой группы (мы их будем называть конечными методами) обеспечивают решение задачи за конечное число шагов. Методы второй группы (итеративные методы) связаны с бесконечным числом итераций и позволяют получить, вообще говоря, лишь приближенное решение задачи. При этом качество приближения существенно зависит от числа проведенных итераций.

Конечные методы, в свою очередь, целесообразно разделить на три класса.

1. Методы, связанные с последовательным улучшением плана.

2. Методы, связанные с последовательным уточнением оценок.

3. Методы, связанные с последовательным сокращением невязок.

Смысл введенной терминологии станет ясен из следующего.

Ниже мы приведем краткое описание различных классов конечных и итеративных методов, выделяя в каждом классе наиболее характерные методы. Опубликованные в литературе общие методы линейного программирования в основном совпадают с описанными здесь методами либо являются теми или иными их модификациями, отличающимися главным образом реализацией алгоритма решения задачи.

Геометрическое истолкование всех методов проводится в многомерном пространстве. Однако ничто не мешает читателю относить все рассуждения к трехмерному пространству, где все построения приобретают геометрическую наглядность.

В приложениях линейного программирования достаточно общего характера чаще всего используют различные варианты метода последовательного улучшения плана. Сущность метода может быть изложена следующим образом.

Предполагается наличие некоторого исходного опорного плана. В основе метода лежит определенный критерий проверки опорного плана на оптимальность. Если исходный опорный план оказывается неоптимальным, метод рекомендует путь улучшения плана. Указывается способ перехода к другому опорному плану, более близкому к оптимальному. Метод гарантирует, что таким образом, постепенно улучшая план, можно через конечное число шагов достигнуть оптимального плана — решения задачи линейного программирования.

Метод имеет простое геометрическое истолкование. Опорный план, как мы уже видели, соответствует вершине многогранника условий. Решение задачи линейного программирования по методу последовательного улучшения плана начинается, таким образом, с рассмотрения некоторой вершины многогранника условий. Для того чтобы проверить, достигается ли экстремум линейной формы в исходной вершине, достаточно сравнить значение линейной формы в этой вершине и во всех соседних с ней *). Аналитическая запись критерия оптимальности как раз и отражает сравнение значений линейной формы в соседних вершинах многогранника условий. Если исходная вершина не соответствует оптимуму линейной формы, то осуществляется переход к соседней вершине, увеличивающей значение формы при решении задач на максимум и уменьшающей ее при решении задач на минимум. Многогранник условий имеет конечное число вершин, а каждый переход от одной вершины к соседней изменяет значение линейной формы в одну и

*) Соседними вершинами называются вершины, принадлежащие одному и тому же ребру многогранника. Ребро многогранника в n -мерном пространстве получается пересечением $(n - 1)$ линейно независимых гиперплоскостей.

ту же сторону. Поэтому невозможно оказаться дважды в одной и той же вершине. Таким образом, гарантируется переход из исходной вершины в вершину, отвечающую оптимуму линейной формы, за конечное число шагов. Большинство вершин при этом не рассматривается. Представляет интерес и вторая геометрическая интерпретация метода последовательного улучшения плана. Исходный план соответствует системе линейно независимых ребер выпуклого конуса (§ 7). Гиперплоскость, натянутая на эти ребра, пересекает прямую Λ в некоторой точке, принадлежащей отрезку, общему для прямой Λ и выпуклого многогранного конуса. Если эта точка лежит на границе конуса, то она соответствует максимуму или минимуму линейной формы. Пусть при вычислении максимума линейной формы точка пересечения оси и гиперплоскости не оказалась «верхней» точкой пересечения оси и конуса. Тогда выбирается ребро конуса, лежащее «выше» гиперплоскости, и вводится в исходную систему ребер вместо одного из них. Новая система ребер соответствует новому опорному плану и определяет новую гиперплоскость. Полученная таким образом гиперплоскость пересекает прямую Λ в точке, расположенной «выше» точки пересечения Λ с предыдущей гиперплоскостью. Так, двигаясь шаг за шагом, можно в конечное число этапов дойти до «верхней» точки пересечения прямой и многогранного конуса и, следовательно, получить решение задачи линейного программирования.

9.3. Второй класс методов решения задач линейного программирования связан с так называемой проблемой двойственности и сопряженной задачей.

Приведем относительно частную формулировку проблемы двойственности. Этого будет достаточно для характеристики особенностей метода последовательного уточнения оценок.

Исходной или прямой задачей линейного программирования является следующая задача.

Требуется найти максимум линейной формы

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n$$

при m условиях вида

$$a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1,$$

в точке $X = (x_1, \dots, x_n)$ всегда не меньше (если в исходной задаче вычисляется максимум формы) максимального значения L . Если все x_i оказываются неотрицательными, то на них достигается экстремум линейной формы L . Другими словами, если коэффициенты разложения x_i оказываются планом исходной задачи, то этот план оптимальный. Если же среди компонент X имеются отрицательные числа, то метод последовательного улучшения плана для сопряженной задачи указывает способ перехода к следующему опорному плану. Очередному плану сопряженной задачи соответствует новый набор чисел $X = (x_1, \dots, x_n)$, который уменьшает значение линейной формы L , приближая ее к искомому оптимуму.

В гл. 4 будет показано, что последовательные опорные планы сопряженной задачи могут быть интерпретированы как последовательные уточнения оценок ограничений исходной задачи. Оптимальный план сопряженной задачи позволяет оценить, насколько сказывается вариация тех или иных условий исходной задачи на величину линейной формы. Отсюда и название метода — метод последовательного уточнения оценок.

Порядок решения задач линейного программирования по методу последовательного уточнения оценок может быть геометрически описан следующим образом.

Каждый опорный план сопряженной задачи во второй геометрической интерпретации задач линейного программирования (§ 7) определяет гиперплоскость H_1 , которая:

- а) проходит через начало координат;
- б) расположена «выше» многогранного конуса;
- в) содержит m -мерную грань конуса.

Точка пересечения гиперплоскости H_1 с прямой Λ расположена «выше» конуса или на его поверхности. Если точка пересечения H_1 и Λ оказывается на поверхности конуса, то она является крайней точкой пересечения оси и конуса и определяет решение задачи. Если точка пересечения H_1 и Λ оказывается выше конуса, то метод дает правило перехода к следующему приближению. Указывается, какое из ребер конуса следует исключить из системы лучей, определяющих гиперплоскость H_1 . Затем гиперплоскость поворачивается относительно остальных ребер до тех пор, пока она не совместится с другой гранью выпуклого многогранного конуса. Поворот осу-

ществляется так, что полученная плоскость H_2 , соответствующая новому опорному плану сопряженной задачи, пересекает прямую Λ «ниже», чем H_1 . Продолжая таким образом переход от одной гиперплоскости, содержащей грань многогранного конуса, к следующей и снижая на каждом шаге значение линейной формы L , придем через конечное число шагов к решению задачи линейного программирования.

Как мы видели, метод последовательного улучшения плана позволяет приближаться к оптимальному значению линейной формы изнутри конуса, а метод последовательного уточнения оценок извне. Комбинируя оба метода, можно получить удобный приближенный метод решения задач линейного программирования, требующий обычно относительно небольшого числа итераций. При достижении приемлемого малого значения разности между приближениями, полученными по обоим методам, можно прекратить расчет. В практических задачах не всегда можно рассчитывать на точное знание коэффициентов линейной формы и параметров условий. В таких случаях целесообразно допустить сокращение трудоемкости расчетов в ущерб точности решения задачи.

Приближенный метод требует решения двух задач — прямой и сопряженной. Поэтому трудоемкость каждого шага удваивается. Однако число шагов может при этом существенно сократиться. По достигнутой разности значений линейных форм двойственных задач можно после каждого шага судить о необходимости продолжения вычислений.

9.4. Характерным представителем третьего класса методов является описанный в гл. 4 метод последовательного сокращения невязок.

Если методы первого класса были связаны с рассмотрением исходной задачи, а методы второго класса были основаны на решении сопряженной задачи, то методы третьего класса требуют рассмотрения прямой и сопряженной задач в комплексе.

Вычисления начинаются с определения некоторого начального плана сопряженной задачи (не обязательно опорного). Начальному плану известным образом приводится в соответствие некоторая система S векторов условий исходной задачи. Основная операция каждого шага состоит в выборе вектора P — линейной комбинации

векторов системы S с положительными коэффициентами, возможно меньше отличающейся от вектора ограничений. Последнее требование следует понимать таким образом. Компоненты вектора P не должны превышать соответствующих составляющих вектора ограничений, и при этом сумма уклонений компонент вектора P от ограничений b_i (сумма невязок) должна быть минимально возможной. Как видим, вычисление требуемой линейной комбинации векторов условий системы S , в свою очередь, является задачей линейного программирования. Но это упрощенная задача, и решение ее методом последовательного улучшения плана существенно проще, чем решение исходной задачи.

Если начальный план сопряженной задачи выбран так, что все невязки оказались равными нулю, то полученные коэффициенты x_k разложения вектора P по векторам системы S определяют оптимальный план исходной задачи. Для случая, когда среди невязок оказываются ненулевые величины, метод указывает, как при помощи коэффициентов x_k (точнее, некоторых величин, связанных с ними) и предыдущего плана сопряженной задачи построить новый план сопряженной задачи, приводящий к сокращению невязок. Двигаясь таким образом и сокращая шаг за шагом невязки, можно через конечное количество шагов обратить все невязки в нуль и вычислить оптимальный план задачи линейного программирования.

Заметим, что в упрощенной задаче (в гл. 4 она названа вспомогательной задачей) общее количество переменных существенно меньше, чем в исходной задаче. Поэтому решение упрощенной задачи связано с значительно менее трудоемкой работой. Кроме того, оптимальный план упрощенной задачи на предыдущем шаге является планом упрощенной задачи на последующем шаге, обычно близким к ее решению.

Приведем геометрическое истолкование метода последовательного сокращения невязок. Начальный план сопряженной задачи во второй геометрической интерпретации соответствует гиперплоскости H , проходящей через начало координат и расположенной «выше» выпуклого многогранного конуса. Гиперплоскость H может и не содержать грани конуса m -й размерности, поскольку в отличие от метода уточнения оценок здесь не

обязательно отправляться от опорного плана. Это значит, что гиперплоскость H содержит, вообще говоря, менее m ребер конуса. Ребра конуса, принадлежащие H , образуют систему S . Вектор P , которым следует приблизить вектор ограничений по всем его компонентам, должен быть образован как линейная комбинация с положительными коэффициентами ребер системы S . Система ребер S образует грань Q конуса, размерность которой не превышает m . Невязки обращаются в нуль, если грань Q содержит точку пересечения прямой Λ и гиперплоскости H . Гиперплоскость H определяет в этом случае решение сопряженной, а следовательно, и исходной задачи. Если грань Q не содержит точки пересечения H и Λ , то необходимо переходить к следующему приближению. Метод указывает, как по гиперплоскости H и по вектору P , полученному в результате решения упрощенной задачи, построить новую гиперплоскость, сокращающую величины невязок. Конечным числом шагов невязки сводятся к нулю.

В отличие от предыдущих методов движение к оптимальному плану осуществляется здесь по границе конуса, соответствующего задаче.

Рассмотренные выше три метода решения задач линейного программирования будут подробно изучены в последующих главах, где разработка их доводится до вычислительных алгоритмов. Заметим, что совокупность рассмотренных конечных методов является в известном смысле полной. Это следует из геометрической интерпретации методов в $(m+1)$ -мерном пространстве. Первый класс методов связан с движением к оптимуму изнутри конуса задачи. В методах второго класса к решению приближаются извне конуса. Наконец, третий класс методов приводит к движению по границе конуса.

9.5. Итеративные методы вычисления оптимального плана созданы большей частью вне линейного программирования. Многие из них возникли из потребностей теории игр.

Рамки настоящей книги не позволяют сколько-нибудь подробно изложить математические схемы игр и методы их решения. Читателю, знакомому с теорией игр, последующие замечания могут оказаться полезными. Впрочем, некоторые сведения из теории игр можно почерпнуть из § 2 заключения.

Решение прямой и сопряженной задач линейного программирования всегда может быть сведено к вычислению оптимальных смешанных стратегий двух игроков в прямоугольной игре с нулевой суммой. Это означает, что методы решения матричных игр могут быть использованы для решения задач линейного программирования. К сожалению, все известные методы решения игр при немалом числе стратегий у противников оказываются сложнее описанных здесь методов линейного программирования. По-видимому, методы линейного программирования будут чаще использованы для решения игр, чем существующие игровые методы для линейного программирования. Однако имеются основания ожидать существенного улучшения сходимости приближенных методов решения игр. При этом игровые методы могут оказаться полезными для решения широкого класса задач линейного программирования. Это относится, в частности, к методу «вилки», предложенному Брауном [23] и обоснованному Ю. Робинсоном [66].

Метод «вилки» фиксирует произвольную начальную стратегию одного игрока и применительно к ней вычисляет наивыгоднейшую стратегию другого игрока. Затем определяется поведение первого игрока, наилучшее против вычисленной таким образом стратегии противника. Следующий шаг второго игрока является наилучшим ответом против поведения первого игрока на первом и втором его шагах и т. д. В терминах линейного программирования можно дать следующую характеристику этого приближенного метода решения игр. Метод «вилки» связан с одновременным решением прямой и сопряженной задач. Однако рассматриваемые на отдельных этапах планы могут не быть опорными. Поэтому точное решение задачи здесь далеко не всегда может быть достигнуто в конечное число шагов. Сходимость метода обеспечивается, однако монотонность итераций не гарантируется. Тем не менее после каждого шага нетрудно вычислить границы — «вилку», внутри которой заключается оптимальная величина линейной формы.

Каждый шаг в методе «вилки» требует существенно меньшего количества вычислений, чем комбинированное использование метода последовательного улучшения плана и метода последовательного уточнения оценок. Однако для обеспечения той же «вилки» приближенный

игровой метод требует гораздо большего числа шагов.

В § 2 заключения описывается еще один итеративный метод решения игр, принадлежащий фон Нейману, и указываются возможные пути улучшения сходимости метода.

В настоящее время еще нет достаточно точных теоретических результатов, позволяющих сравнить трудоемкость конечных и итеративных методов решения задач линейного программирования. Однако опыт вычислений говорит в пользу конечных методов, реализация которых практически связана с числом шагов, колеблющимся от m до $2m$. (Заметим, что каждый шаг в любом из описанных здесь конечных методов требует порядка mn умножений.) Конечно, можно построить примеры, в которых число итераций существенно больше $2m$. Однако, как правило, практические задачи не приводят к таким примерам. В силу отмеченных здесь обстоятельств в настоящем руководстве основное внимание уделено конечным методам.

9.6. В заключение параграфа приведем несколько замечаний общего характера, позволяющих в ряде случаев отдавать предпочтение тому или иному методу решения задач линейного программирования.

Метод последовательного уточнения оценок и метод сокращения невязок требуют значительно меньше вычислений, чем метод последовательного улучшения плана, если число условий задачи существенно меньше числа переменных ($m \ll n$). В этом случае многогранник условий сопряженной задачи имеет меньше вершин, чем многогранник исходной задачи. Естественно ожидать, что с уменьшением числа вершин уменьшится и трудоемкость их упорядоченного перебора.

При решении вопроса о выборе метода вычисления оптимального плана целесообразно учитывать, кроме того, следующие обстоятельства.

В различных задачах линейного программирования определение начального опорного плана для прямой и сопряженной задач связано с разной трудоемкостью. В тех случаях, когда легко найти опорный план для исходной задачи (не вводя дополнительных переменных), обычно отдают предпочтение методу последовательного улучшения плана. Если легче получить опорный план для сопряженной задачи, то чаще всего трудоемкость

решения задачи по методу последовательного уточнения оценок оказывается меньшей, чем при использовании метода последовательного улучшения плана. В тех случаях, когда неопорный план вычисляется без труда, а определение опорного плана для прямой и для сопряженной задачи вызывает затруднения, следует рекомендовать метод последовательного сокращения невязок. В ряде случаев удается без труда определить исходные опорные планы прямой и сопряженной задач. Обычно физические соображения дают основание судить о том, какой из них ближе к оптимальному плану соответствующей задачи. В зависимости от этого и выбирается метод решения задачи линейного программирования.

§ 10. КРАТКАЯ ИСТОРИЧЕСКАЯ СПРАВКА

Линейное программирование — новая бурно развивающаяся отрасль математики — возникла из практических потребностей несколько более двух десятков лет тому назад.

Отдельные работы, касающиеся частных вопросов линейного программирования, относятся еще к началу 30-х годов. Например, в 1931 г. в Венгрии была опубликована статья Эгервари [38], посвященная проблеме выбора — частному случаю транспортной задачи. Интересно заметить, что в дальнейшем на основе результатов этой статьи рядом авторов был разработан весьма эффективный метод решения транспортной задачи [51], [52], [60], [41]. В литературе этот метод получил название венгерского.

Термин «линейное программирование» появился в Америке в середине 40-х годов (первая американская работа по частной задаче линейного программирования опубликована в 1941 г. [48]). Однако родиной линейного программирования можно по праву считать Советский Союз. В конце 30-х годов целый ряд существенных результатов по линейному программированию был установлен Л. В. Канторовичем [6], [7], [11] *). В частности, в работе [6] была высказана и применена к ряду частных задач чрезвычайно плодотворная идея — идея разре-

*) Работа [11], опубликованная в 1949 г., была выполнена в 1940 г. Ее выходу в свет помешала война. В статье [8], относящейся к 1942 г., имеется ссылка на эту работу.

шающих множителей, тесно связанная с проблемой двойственности в линейном программировании.

В терминах разрешающих множителей весьма удобно формулировать условия оптимальности линейных задач. Совокупность алгоритмов, использующих разрешающие множители, иногда объединяют одним названием — метод разрешающих множителей. В работе [6] была намечена одна из реализаций метода разрешающих множителей, связанная с движением по границе конуса исследуемой задачи (метод последовательного сокращения невязок). В той же работе указана геометрическая интерпретация задач линейного программирования. Геометрическое описание метода сокращения невязок было дано в работах Г. Ш. Рубинштейна [13], [14].

В 1956 г. Данциг, Форд и Фулкерсон [3] на основе упомянутого выше венгерского метода разработали общий метод линейного программирования, который лишь в незначительных деталях отличается от метода, предложенного Л. В. Канторовичем. В совместной работе Л. В. Канторовича и М. К. Гавурина [11] была предложена применительно к транспортной задаче еще одна реализация метода разрешающих множителей, связанная с движением изнутри конуса задачи (метод потенциалов).

Первая американская работа по общим вопросам линейного программирования опубликована Данцигом в 1949 г. [28]. В ней изложены основные идеи метода последовательного улучшения плана (I алгоритм) применительно к невырожденным задачам линейного программирования.

В дальнейшем Чарнес [25] указал прием, позволяющий применять метод улучшения плана и в вырожденных случаях. Метод улучшения плана, связанные с ним вычислительные алгоритмы и различные модификации метода рассматривались Данцигом и рядом других авторов ([29], [36], [37], [56], [62] и [69]).

В 1954 г. Лемке [54] был предложен еще один общий метод линейного программирования, сводящийся к движению извне конуса задачи (метод последовательного уточнения оценок). Известны также такие общие методы линейного программирования, как градиентный [20] и близкий к нему метод логарифмического потенциала [42] (в настоящей книге эти методы не рассмат-

риваются). Параллельно с общими методами линейного программирования создавались методы решения различных частных задач, таких, как транспортная задача, проблема выбора и т. д. Наиболее важные результаты в этой области получены в работах [38], [30], [2] и [51].

Качественным результатам в теории линейного программирования (проблеме двойственности и др.) и связи линейного программирования с теорией игр посвящены работы [31], [32], [35], [44] и [43]. Очень интересными и важными являются новые постановки в линейном программировании, связанные со случайными ([65]) и бесконечными ([4], [63], [10]) программами.

В ряде работ методы линейного программирования применяются к построению чебышевских многочленов наилучшего приближения.

В некоторых случаях нелинейные экстремальные задачи могут быть решены методами линейного программирования [27]. Несмотря на то, что для теории решения нелинейных экстремальных задач уже придуман особый термин «нелинейное программирование», литература по этим вопросам крайне бедна ([53], [68]).

Следует отметить, что за границей работы советских математиков, посвященные линейному программированию, долгое время оставались неизвестными. В связи с этим целый ряд результатов линейного программирования «переоткрывался» иностранными учеными. В последнее время приоритет советских математиков в линейном программировании отмечается не только у нас, но и за рубежом.

Линейное программирование является новой быстро развивающейся наукой. Результаты, методы и алгоритмы линейного программирования зачастую разбросаны по различным журнальным статьям. Часто новые направления в этой науке намечаются не в математических работах, а в прикладных изданиях. Поэтому приведенную здесь историческую справку нельзя считать достаточно полной. Читателю, желающему получить более полное представление об иностранной литературе по линейному программированию, следует обратиться к библиографии В. Роде [67], а также к аннотированным библиографиям [64], [21]. Достаточно полная библиография содержится в недавно вышедшей книге [19].

ГЛАВА 2

ПРАКТИЧЕСКИЕ ЗАДАЧИ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В настоящей главе излагаются постановки ряда народнохозяйственных, технических и военных задач, решение которых сводится к использованию методов линейного программирования.

§ 1. ЗАДАЧА О СМЕСИ

Задача о смеси является одним из первых применений линейного программирования к практическим потребностям. Эта задача в различных формулировках применительно к самым разнообразным приложениям часто обсуждалась в периодической литературе.

В военные годы была поставлена задача о диете. Сущность ее в следующем. Различные пищевые продукты содержат питательные вещества и витамины необходимых типов в определенных, но различных пропорциях. Минимальные потребности в тех или иных питательных веществах и витаминах известны. Зная величину запасов различных продуктов и стоимость каждого из них, необходимо определить, как можно удовлетворить потребности населения при минимальных затратах.

Приведем математическую формулировку той же задачи.

Пусть имеется n различных продуктов $P_1, P_2, \dots, P_n$. Обозначим через x_i суточное потребление i -го продукта. Диета характеризуется, таким образом, системой чисел $X = (x_1, x_2, \dots, x_n)$.

С каждой диетой связывается соответствующее ей количество питательных веществ и микроэлементов — бел-

ков, жиров, углеводов, железа, кальция, витаминов. Пусть величина a_{ij} характеризует содержание j -го питательного вещества в единице i -го продукта. Тогда общее содержание j -го питательного вещества в принятой диете равно

$$a_{1j}x_1 + a_{2j}x_2 + \dots + a_{nj}x_n,$$

$j = 1, 2, \dots, m$; m — общее число требуемых питательных веществ.

Обозначим минимальную ежесуточную потребность организма в j -м питательном веществе через b_j . Это означает, что диета должна удовлетворять ограничениям

$$a_{1j}x_1 + a_{2j}x_2 + \dots + a_{nj}x_n \geq b_j \quad (2.1)$$

$$(j = 1, 2, \dots, m).$$

С другой стороны, суточное потребление каждого продукта ограничено запасами, а также возможностями и сроками их пополнения. Отсюда дополнительные условия на компоненты диеты

$$0 \leq x_i \leq a_i, \quad (2.2)$$

$$(i = 1, 2, \dots, n),$$

где величины a_i определяются наличными ресурсами продуктов Π_i и временем, на которое они рассчитаны.

Если c_i — стоимость единицы i -го продукта, то стоимость всей диеты определяется линейной формой

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n. \quad (2.3)$$

Оптимальной диетой следует признать такую систему ежесуточного потребления продуктов $X = \{x_1, x_2, \dots, x_n\}$, которая обращает в минимум L (2.3) при соблюдении условий (2.1) и (2.2).

В ряде практических задач на составление диеты необходимо также учитывать ограничения сверху на потребление некоторых питательных веществ, т. е. вместо условий (2.1) следует рассматривать ограничения вида

$$\bar{b}_j \geq a_{1j}x_1 + a_{2j}x_2 + \dots + a_{nj}x_n \geq b_j$$

$$(j = 1, 2, \dots, m).$$

Другой более сложный вариант задачи о смеси — это задача о получении оптимального авиационного горючего [26].

Имеется n компонент горючего, различающихся своими характеристиками и расходами, связанными с их получением и хранением. Смешивая составляющие в разных пропорциях, меняют качество и стоимость горючего.

Требуется при наименьших затратах получить горючее, удовлетворяющее заданным требованиям по характеристикам летучести и зажигания. При этом необходимо учитывать ограниченные запасы некоторых составляющих.

Ясно, что к подобным схемам может быть сведено большое число задач о составлении оптимальных рецептов, сплавов и смесей, удовлетворяющих заранее заданным требованиям.

§ 2. ОБ ОПТИМАЛЬНОМ РАСКРОЕ МАТЕРИАЛОВ

Одна из первых прикладных задач, потребовавших упорядочения перебора вариантов — это так называемая «задача фанерного треста» [6]. По-видимому, именно эта задача положила начало развитию методов линейного программирования в Советском Союзе. Задача фанерного треста в дальнейшем не раз обобщалась и может быть сформулирована как задача о максимальном сокращении отходов или задача об оптимальном раскрое материалов.

Полуфабрикаты поступают на предприятие в виде листов фанеры, жести, стали, стекла. Из полуфабрикатов требуется изготовить возможно большее число комплектов деталей. При этом должны быть соблюдены следующие условия.

Всего имеется n партий материала, причем i -я партия содержит q_i единиц. Комплект состоит из m разных деталей. В комплект входит p_k деталей k -го типа. Единица каждой партии может раскраиваться s различными способами.

Пусть при j -м способе раскроя i -й партии полуфабрикатов получается a_{ikj} деталей k -го типа.

Обозначим через x_{ij} количество единиц i -й партии материалов, которые следует раскроить по j -му способу.

Количество деталей k -го типа, которое будет при этом получено, равно $a_{ikj}x_{ij}$. Количество деталей k -го типа, которое можно получить из i -й партии полуфабрикатов, используя все способы раскроя, равно

$$a_{ik1}x_{i1} + a_{ik2}x_{i2} + \dots + a_{iks}x_{is} = \sum_{j=1}^s a_{ikj}x_{ij}.$$

Общее число деталей k -го типа, очевидно, можно получить, если сложить количества деталей этого вида, выкраиваемых из каждой партии материалов:

$$\sum_{j=1}^s a_{1kj} x_{1j} + \sum_{j=1}^s a_{2kj} x_{2j} + \dots + \sum_{j=1}^s a_{nkj} x_{nj} = \sum_{i=1}^n \sum_{j=1}^s a_{ikj} x_{ij}.$$

Каждый комплект содержит p_k деталей k -го типа. Поэтому количество комплектов, для которых заготовлено деталей k -го типа, равно

$$Z_k = \frac{\sum_{i=1}^n \sum_{j=1}^s a_{ikj} x_{ij}}{p_k}.$$

Комплект должен быть обеспечен деталями всех типов. Это значит, что задача об оптимальном раскрое материалов сводится к вычислению набора чисел x_{ij} ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, s$), при котором обеспечивается максимум минимального из отношений z_k ($k = 1, 2, \dots, m$). Другими словами, необходимо обеспечить максимум z при условиях

$$z_k \geq z \quad (k=1, 2, \dots, m).$$

Кроме того, должны быть учтены условия

$$\left. \begin{aligned} x_{11} + x_{12} + \dots + x_{1s} &= q_1, \\ x_{21} + x_{22} + \dots + x_{2s} &= q_2, \\ . &. \\ x_{n1} + x_{n2} + \dots + x_{ns} &= q_n, \end{aligned} \right\}$$

$$x_{ij} \geq 0 \quad (i=1, 2, \dots, n; j=1, 2, \dots, s).$$

В первой группе условий записано, что i -я партия содержит q_i единиц материала. Вторая группа неравенств не требует пояснения.

Естественно еще потребовать, чтобы переменные x_{ij} были целыми числами. Однако в общем случае неясно, какие дополнительные условия следует ввести, чтобы гарантировать целочисленные решения задачи. В тех случаях, когда $x_{ij} \gg 1$ или $x_{ij} \ll 1$, округление дробных решений до ближайших целых чисел, как правило, не вызывает заметного уменьшения числа комплектов. Некоторые соображения о решении задач линейного программирования в целых числах приведены в § 1 заключения.

Приведем сформулированную задачу линейного программирования к каноническому виду. Введем дополнительные переменные $\zeta_k \geq 0$ ($k=1, 2, \dots, m$).

Задача об оптимальном раскрое может быть теперь записана следующим образом.

Требуется обратить в максимум линейную форму

$$L = z_1 - \zeta_1 = \frac{1}{p_1} \left\{ \sum_{i=1}^n \sum_{j=1}^s a_{i1j} x_{ij} - p_1 \zeta_1 \right\} \quad (2.4)$$

при условиях

$$\sum_{i=1}^n \sum_{j=1}^s a_{ikj} x_{ij} - p_k \zeta_k = p_k L \quad (k=2, 3, \dots, m),$$

$$\sum_{j=1}^s x_{ij} = q_i \quad (i=1, 2, \dots, n),$$

$$x_{ij} \geq 0 \quad (i=1, 2, \dots, n; j=1, 2, \dots, s),$$

$$\zeta_k \geq 0 \quad (k=1, 2, \dots, m).$$

Обозначим через A_{ij} вектор условий при переменной x_{ij} ; B , как обычно, вектор ограничений.

Векторы условий для переменных x_{ij} и вектор ограничений задачи об оптимальном раскрое имеют следующий вид:

$$A_{ij} = \left\{ a_{i2j} - \pi_2 a_{i1j}, \dots, a_{imj} - \pi_m a_{i1j}, \underbrace{0, \dots, 0, 1, 0, \dots, 0}_i \right\},$$

где $\pi_s = \frac{p_s}{p_1}$;

$$B = \left\{ \underbrace{0, \dots, 0}_{m-1}, q_1, q_2, \dots, q_n \right\}.$$

§ 3. РАСПРЕДЕЛЕНИЕ САМОЛЕТОВ МЕЖДУ ВОЗДУШНЫМИ ЛИНИЯМИ *)

Имеется n типов самолетов, которые должны быть использованы для перевозки пассажиров по m линиям. Всего подлежит распределению N самолетов. При этом число самолетов i -го типа равно N_i . Очевидно, $\sum_{i=1}^n N_i = N$.

Самолеты разных типов при эксплуатации на той или иной воздушной линии имеют различные характеристики. Из-за недостаточной дальности полета некоторые самолеты не могут быть эффективно использованы на линиях большой протяженности. Самолеты разных типов имеют различное число посадок в пути. На некоторых линиях не используется полностью коммерческая грузоподъемность самолетов. Все это приводит к тому, что месячный объем перевозок и расходов на один самолет различен для разных типов самолетов и разных воздушных линий.

Исходя из данных о себестоимости пассажирокилометра и коммерческой загрузки каждого типа самолетов на каждой авиалинии, устанавливаются:

1°. Величины a_{ij} — месячный объем перевозок пассажиров одним самолетом i -го типа по j -й воздушной линии.

2°. c_{ij} — месячный расход на один самолет i -го типа, эксплуатируемый на j -й авиалинии.

Предполагается также известным число пассажиров b_j , подлежащих перевозке в месяц по j -й линии.

Задача заключается в распределении самолетного парка по m авиалиниям для перевозки заданного количества пассажиров при минимальных затратах.

Обозначим через x_{ij} количество самолетов i -го типа на j -й воздушной линии.

Суммарная величина расходов записывается в виде линейной формы

$$L = \sum_{i=1}^n \sum_{j=1}^m c_{ij} x_{ij}. \quad (2.5)$$

*) Задача подобного типа рассматривалась в [39].

Значение L требуется минимизировать. Условия, при которых должна решаться задача, следующие.

Самолеты должны быть распределены по авиалиниям, чтобы обеспечить перевозку по j -й линии не менее b_j пассажиров. Формально эти условия записываются в виде системы неравенств

$$\sum_{i=1}^n a_{ij} x_{ij} \geq b_j \quad (j=1, 2, \dots, m). \quad (2.6)$$

Общее количество самолетов i -го типа, подлежащих распределению, равно по условию N_i . Следовательно,

$$\sum_{j=1}^m x_{ij} = N_i \quad (i=1, 2, \dots, n). \quad (2.7)$$

По физическому смыслу x_{ij} очевидно

$$x_{ij} \geq 0 \quad (i=1, 2, \dots, n; j=1, 2, \dots, m).$$

Кроме того, x_{ij} должны быть целыми числами. Вопросу о целочисленных решениях таких задач посвящен п. 1.2 заключения (гл. 7).

Векторы условий A_{ij} задачи линейного программирования, к которой мы пришли, содержат по $m+n$ компонент, из которых только две отличны от нуля — это a_{ij} на j -м месте и 1 — на $(m+i)$ -м. Составляющие вектора ограничений — это величины b_i и N_i .

$$A_{ij} = \left\{ \underbrace{0, \dots, 0}_j, \overbrace{a_{ij}, 0, \dots, 0}^{m+i}, 1, 0, \dots, 0 \right\},$$

$$B = \{b_1, b_2, \dots, b_m, N_1, N_2, \dots, N_n\}.$$

§ 4. СЕЛЬСКОХОЗЯЙСТВЕННЫЕ ЗАДАЧИ

Целый ряд задач сельскохозяйственной практики решается методами линейного программирования. Сюда относятся задачи о наивыгоднейшем распределении посевной площади, о составлении рациона откорма скота и птицы, о графике работы сельскохозяйственных машин и т. д. Начнем с рассмотрения следующей задачи [6]. В состав района входит n колхозов, посевные пло-

щади которых равны соответственно $s_1, s_2, \dots, s_n$. Согласно плану в районе должно производиться m культур в соотношении $p_1 : p_2 : p_3 : \dots : p_m$. Различие почв и климатических условий определяет различную пригодность отдельных участков для выращивания разных сельскохозяйственных культур. Пусть ожидаемый урожай j -й культуры в i -м колхозе с одного гектара площади равен a_{ij} .

Необходимо выяснить, какую часть посевной площади каждый колхоз должен отвести под ту или иную культуру, чтобы обеспечить максимальный урожай и сохранить предписанное соотношение культур.

Пусть x_{ij} — число гектаров, которое i -й колхоз должен отвести под j -ю культуру. Тогда

$$\sum_{j=1}^m x_{ij} = s_i \quad (i = 1, 2, \dots, n) \quad (2.8)$$

— общая посевная площадь i -го колхоза.

Ожидаемый урожай j -й культуры со всей посевной площади района равен

$$z_j = \sum_{i=1}^n a_{ij} x_{ij}.$$

Задача об оптимальном распределении посевных площадей сводится к такому подбору системы чисел $x_{ij} \geq 0$, при котором имеет место условие (2.8), а система отношений

$$L = \frac{z_1}{p_1} = \frac{z_2}{p_2} = \dots = \frac{z_m}{p_m}$$

достигает максимума.

Если земля поливная и c_{ij} — норма расхода воды для j -й культуры на участке i -го колхоза, то дополнительные условия записываются в виде

$$\sum_{j=1}^m c_{ij} x_{ij} \leq c_i \quad (i = 1, 2, \dots, n),$$

где c_i — производительность источников орошения в i -м колхозе.

Точно таким же образом записываются условия, ограничивающие наличие или возможность применения техники и рабочей силы для обработки разных культур в различных колхозах.

Задача приводится к каноническому виду так же, как и задача об оптимальном раскрое материала (§ 2).

К задаче рассмотренного вида сводится и задача распределения механизмов и машин между различными участками работы. Суммарное время простоя механизмов должно сводиться к минимуму. При этом необходимо учитывать ограничения по горючему, по обслуживающему персоналу и т. д.

Задача о диете, рассмотренная в § 1, представляет интерес для сельскохозяйственной практики. Ее можно использовать для составления наиболее выгодного рациона откорма скота.

В ряде случаев к задаче о распределении посевной площади приходится подходить из соображений, отличных от приведенных выше. Во вновь осваиваемых районах, особенно на целине, равномерная загрузка рабочей силы в течение года является весьма важным фактором. Часто этот фактор оказывается решающим для закрепления населения за определенными районами. Проводимые обычно мероприятия, такие, как перенос отпусков на зиму, строительство ремонтных мастерских, консервных заводов и т. д., не решают полностью проблему равномерной загрузки основной массы рабочей силы. Поэтому при планировании сельскохозяйственных работ в совхозах необходимо известную часть посевной площади отводить под культуры, главным образом технические, требующие затраты труда в зимнее время года.

Пусть требуется распределить площадь под n культур. Разобьем год на m периодов, не обязательно одинаковой продолжительности. Естественно, например, май считать за 3 периода, а всю зиму за один период.

Введем следующие обозначения:

- x_i — площадь, отведенная под i -ю культуру,
- b_j — количество рабочей силы, которой располагает совхоз в j -й период,
- d_j — число людей, которое может быть в j -й период занято не сельскохозяйственными работами.

a_{ij} — число людей, которое следует занять в j -й период на i -й культуре, если под нее отведена единица посевной площади,

c_i — прибыль от i -й культуры, приходящаяся на единицу посевной площади.

Общие доходы совхоза равны

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n.$$

Наивыгоднейшее распределение посевной площади определяется здесь системой чисел $\{x_1, x_2, \dots, x_n\}$, обращающих линейную форму в максимум при условии:

$$1^\circ. \quad x_i \geq \frac{a_i}{\alpha_i} \quad (i = 1, 2, \dots, n),$$

где a_i — плановое задание по производству i -й культуры (некоторые a_i могут быть равны нулю);
 α_i — урожайность i -й культуры.

$$2^\circ. \quad b_j - d_j \leq \sum_{i=1}^n a_{ij}x_i \leq b_j \quad (j = 1, 2, \dots, m).$$

Последнее условие гарантирует равномерную загрузку рабочей силы в течение года.

$$3^\circ. \quad x_1 + x_2 + \dots + x_n \leq a,$$

где a — посевная площадь совхоза.

Чтобы перейти к канонической форме задачи линейного программирования, удобно ввести замену

$$y_i = x_i - \frac{a_i}{\alpha_i}$$

и переписать условие задачи следующим образом.

Требуется обратить в максимум линейную форму

$$\bar{L} = \sum_{i=1}^n c_i y_i$$

при условиях

$$\sum_{i=1}^n a_{ij}y_i + \xi_j = \tilde{b}_j; \quad \sum_{i=1}^n a_{ij}y_i - \eta_j = \tilde{b}_j - d_j,$$

$$\sum_{i=1}^n y_i + \zeta = a - \tilde{a},$$

где

$$\tilde{b}_j = b_j - \sum_{i=1}^n \frac{a_{ij} a_i}{\alpha_i}, \quad \tilde{a} = \sum_{i=1}^n \frac{a_i}{\alpha_i};$$

ξ_j , η_j и ζ — дополнительные переменные,
 $y_i \geq 0$, $\xi_j \geq 0$, $\eta_j \geq 0$, $\zeta \geq 0$ ($i = 1, 2, \dots, n$; $j = 1, 2, \dots, m$).

Векторы условий, соответствующие основным переменным, и вектор ограничений имеют здесь следующий вид:

$$A_i = \{a_{i1}, a_{i2}, \dots, a_{im}, a_{i1}, a_{i2}, \dots, a_{im}, 1\};$$

$$B = \{\tilde{b}_1, \tilde{b}_2, \dots, \tilde{b}_m, \tilde{b}_1 - d_1, \tilde{b}_2 - d_2, \dots, \tilde{b}_m - d_m, a - \tilde{a}\}.$$

§ 5. ЗАДАЧИ О РАЗМЕЩЕНИИ ОБОРУДОВАНИЯ

Ряд практических задач, таких, как размещение оборудования на самолете, компактная укладка разнородного оборудования и т. д., сводится к следующей схеме.

Задано n предметов, каждый из которых применительно к рассматриваемому приложению характеризуется определенным коэффициентом полезности. В различных конкретных задачах коэффициент полезности может описывать различные качества предметов. В одних случаях — это стоимость, в других — калорийность, в третьих — эффективность вооружения и т. д. Единственное требование к коэффициенту полезности заключается в том, чтобы общая полезность размещенного оборудования определялась суммой коэффициентов полезности размещенных предметов. Так, общая стоимость уложенного оборудования составляется из цен отдельных предметов, общая теплотворная способность перевозимого горючего — линейная комбинация калорийностей отдельных видов топлива, математическое ожидание числа пораженных целей установленными видами вооружения представляется как сумма характеристик эффективности каждого типа оружия и т. д.

Все n предметов не могут быть размещены в укладках заданного объема. Кроме того, допустимая грузоподъемность укладок меньше суммарного веса предметов.

Необходимо, соблюдая ограничения по весу и габаритам, обеспечить такое размещение предметов, при котором суммарная полезность уложенного оборудования была бы максимальной.

Часто вместо ограничений по объему приходится соблюдать ограничения по отдельным размерам. Ограничений становится еще больше, если укладка фигурная и размещаемые предметы имеют сложную конфигурацию.

Пусть x_i равно единице, если i -й предмет принимается к укладке, и x_i равно нулю, если этот предмет не выбран.

Введем еще следующие обозначения:

- c_i — коэффициент полезности i -го предмета,
- a_{ij} — j -я характеристика i -го предмета (например, вес, объем или отдельные размеры),
- b_j — ограничение по j -й характеристике.

В принятых обозначениях задача о рациональной укладке оборудования сводится к обращению в максимум линейной формы

$$L = c_1x_1 + c_2x_2 + \dots + c_nx_n = \sum_{i=1}^n c_ix_i$$

при условиях

$$\sum_{i=1}^n a_{ij}x_i \leq b_j,$$

$$0 \leq x_i \leq 1 \tag{2.9}$$

$$(i=1, 2, \dots, n; j=1, 2, \dots, m;$$

m — количество характеристик по которым ограничивается выбор предмета).

Условия (2.9), вообще говоря, не являются достаточными для того, чтобы обеспечить равенство компонент решения x_i нулю или единице. В качестве составляющих решения рассматриваемой задачи могут, вообще говоря,

оказаться и дробные значения x_i . Эти решения обычно не имеют физического смысла.

В предыдущих параграфах уже отмечалось, что округление дробных компонент плана до ближайших целых часто дает удовлетворительное решение задачи. В тех случаях, когда веса и габариты отдельных предметов намного меньше соответствующих ограничений, приближенное решение, полученное округлением, обычно не приводит к заметному уменьшению значения линейной формы по сравнению с ее оптимально достижимой величиной. Однако бывают и случаи, когда округление приводит к решениям, далеким от лучших. Имеются задачи линейного программирования, где нет необходимости в дополнительных условиях для того, чтобы была обеспечена целочисленность решения. К таким задачам относятся, например, подробно исследованные в гл. V транспортная задача и проблема выбора. В общем случае, однако, дополнительные ограничения необходимы. Общий подход к определению дополнительных условий указан в п. 1.2 заключения (гл. 7).

В каждой отдельной задаче физический смысл условий обычно подсказывает, какие дополнительные ограничения необходимо ввести, чтобы обеспечить целочисленность решения. Пусть, например, решение задачи о размещении привело к дробной экстремальной точке

$$\left. \begin{aligned} x_{i_1} &= x_{i_2} = \dots = x_{i_k} = 1, \\ 0 < x_{i_{k+1}} &< 1, \\ x_i &= 0 \text{ для всех остальных } i. \end{aligned} \right\} \quad (2.10)$$

Наличие такого решения означает, что если принять

$$x_{i_1} = x_{i_2} = \dots = x_{i_k} = x_{i_{k+1}} = 1,$$

а остальные x_i равными нулю, то не будет удовлетворяться некоторое ограничение, например ограничение по весу. Ясно, что введение дополнительного условия

$$x_{i_1} + x_{i_2} + \dots + x_{i_k} + x_{i_{k+1}} \leq k$$

приводит к исключению дробной экстремальной точки (2.10). Можно показать, что все целочисленные планы задачи удовлетворяют дополнительному условию.

Может оказаться, что при решении задачи с дополнительным ограничением снова получится дробная экстремальная точка. В этом случае следует опять подбирать условие, исключающее ее. Обычно это условие сводится к ограничению другой частной суммы x_{is} . Однако иногда могут потребоваться и другие, более искусственные соотношения.

§ 6. ОБЩАЯ ПЛАНОВО-ПРОИЗВОДСТВЕННАЯ ЗАДАЧА

В 1957 г. в «Докладах Академии Наук СССР» была опубликована статья Л. В. Канторовича [9], существенно обобщающая производственную задачу, рассмотренную в § 8 главы I, где приводится экономическая интерпретация задачи линейного программирования.

Производство или группа предприятий должны производить r различных продуктов в заранее заданной пропорции (или, как говорят обычно, с учетом заданного ассортимента). Выпуск продукции требует затрат рабочей силы различной квалификации, сырья, энергии и других производственных факторов. Производственные факторы могут быть использованы не только для производства конечных продуктов, но и для производства вспомогательного оборудования, инструмента и других промежуточных продуктов. При планировании необходимо учитывать наличные ресурсы производственных факторов, ограничивающие их затраты. Предполагается, что к началу работы оборудование или полуфабрикаты, именуемые здесь промежуточными продуктами, отсутствовали и, следовательно, затраты промежуточных продуктов в реальном плане были исключены.

При составлении плана исходят из того, что имеется некоторое количество (N) заранее отработанных способов производства, каждый из которых характеризуется тремя векторами-наборами чисел, определяющими объемы производства каждого конечного продукта, отдельных промежуточных продуктов и затраты по каждому производственному фактору.

Исходные способы производства учитывают главным образом технологические возможности и особенности производства и вовсе не связаны с ограничениями и требованиями в каждом отдельном случае. Комбинируя

различные способы производства, можно составлять планы, удовлетворяющие конкретным условиям производства, учитывающие наличные ресурсы и обеспечивающие требуемый ассортимент. Из всех этих планов необходимо выбрать лучший — план, гарантирующий наиболее высокий уровень производства.

План, таким образом, представляет собой набор чисел, показывающих, с каким удельным весом следует использовать тот или иной из исходных способов производства.

Переведем задачу о планировании производства на формальный язык.

Пусть для производства r конечных продуктов требуется n видов производственных факторов. При этом могут быть получены m промежуточных продуктов.

Обозначим векторы, характеризующие соответственно производство окончательных продуктов, промежуточных продуктов и затраты производственных факторов при s -м исходном способе производства через $A^{(s)}$, $B^{(s)}$ и $C^{(s)}$

$$A^{(s)} = (a_{1s}, a_{2s}, \dots, a_{rs}),$$

$$B^{(s)} = (b_{1s}, b_{2s}, \dots, b_{ms}),$$

$$C^{(s)} = (c_{1s}, c_{2s}, \dots, c_{ns}),$$

$$s = 1, 2, \dots, N.$$

Компоненты векторов показывают объемы производства конечных продуктов, промежуточных продуктов и производственных факторов. Отрицательные компоненты соответствуют затратам.

Каждый план определяется вектором

$$X = (x_1, x_2, \dots, x_N).$$

Плану X соответствует производство i -го окончательного продукта, равное

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{iN}x_N, \quad i = 1, 2, \dots, r.$$

Производство j -го промежуточного продукта при плане X равно

$$b_{j1}x_1 + b_{j2}x_2 + \dots + b_{jN}x_N, \quad j = 1, 2, \dots, m.$$

Затраты k -го производственного фактора при выбранном плане определяются числами

$$c_{k1}x_1 + c_{k2}x_2 + \dots + c_{kN}x_N, \quad k = 1, 2, \dots, n.$$

Пусть наличие производственных факторов задано вектором

$$-C^{(0)} = -(c_{10}, c_{20}, \dots, c_{n0}).$$

Предполагается, что $c_{k0} > 0$. Тогда $-c_{k0}$ определяет допустимые затраты k -го производственного фактора. Запишем теперь перечисленные выше ограничения, которым удовлетворяет допустимый план.

Условия, ограничивающие затраты производственных факторов наличными запасами, записываются в виде

$$c_{k1}x_1 + c_{k2}x_2 + \dots + c_{kN}x_N \geq -c_{k0}, \quad (2.11)$$

$$k = 1, 2, \dots, n.$$

Условия, исключающие затраты промежуточных продуктов, имеют вид

$$b_{j1}x_1 + b_{j2}x_2 + \dots + b_{jN}x_N \geq 0,$$

$$j = 1, 2, \dots, m. \quad (2.12)$$

Остается записать требование ассортиментности продукции.

Пусть комплект продукции состоит из a_{i0} единиц i -го конечного продукта ($i = 1, 2, \dots, r$).

Показатель качества планирования — получение максимального количества k комплектов. При этом, естественно, должны соблюдаться условия (2.11), (2.12) и система ограничений

$$\frac{a_{11}x_1 + \dots + a_{1N}x_N}{a_{10}} \geq k,$$

$$\dots \dots \dots$$

$$\frac{a_{r1}x_1 + \dots + a_{rN}x_N}{a_{r0}} \geq k.$$

Обозначая $\frac{a_{i\lambda}}{a_{i0}}$ через $a_{i\lambda}$ и полагая $k = x_{N+1}$, получим систему соотношений

$$\begin{aligned} a_{i1}x_1 + a_{i2}x_2 + \dots + a_{iN}x_N &\geq x_{N+1} \\ (i = 1, 2, \dots, r). \end{aligned} \quad (2.13)$$

Из физического смысла переменных следует, что x_s должны быть неотрицательны, т. е.

$$x_s \geq 0 \quad (s = 1, 2, \dots, N). \quad (2.14)$$

Задача планирования сводится, таким образом, к определению набора чисел $X = (x_1, x_2, \dots, x_N, x_{N+1})$, при которых соблюдаются условия (2.11) — (2.14), а величина x_{N+1} достигает наибольшего возможного значения. Мы пришли к задаче линейного программирования.

Количество переменных здесь $N + 1$:

$$x_1, x_2, \dots, x_N, x_{N+1}.$$

Линейная форма имеет вид

$$L = x_{N+1},$$

а условия ограничения суть

$$a_{i1}x_1 + a_{i2}x_2 + \dots + a_{iN}x_N - x_{N+1} \geq 0$$

$$(i = 1, 2, \dots, r);$$

$$b_{j1}x_1 + b_{j2}x_2 + \dots + b_{jN}x_N \geq 0$$

$$(j = 1, 2, \dots, m);$$

$$c_{k1}x_1 + c_{k2}x_2 + \dots + c_{kN}x_N \geq -c_{k0}$$

$$(k = 1, 2, \dots, n);$$

$$x_s \geq 0, \quad (s = 1, 2, \dots, N).$$

Выпишем в заключение компоненты векторов условий при $x_1, \dots, x_N, x_{N+1}$ и вектора ограничений

$$A_s = (a_{1s}, \dots, a_{rs}, b_{1s}, \dots, b_{ms}, c_{1s}, \dots, c_{ns}), \\ (s = 1, 2, \dots, N);$$

$$A_{N+1} = (\underbrace{-1, \dots, -1}_r, \underbrace{0, \dots, 0}_{n+m});$$

$$B = (\underbrace{0, \dots, 0}_{r+m}, -c_{10}, \dots, -c_{n0}).$$

Введением дополнительных неотрицательных переменных задача так же, как и в § 2, сводится к канонической форме. При этом переменная x_{N+1} исключается.

§ 7. ПРОБЛЕМА СОСТАВЛЕНИЯ ГРАФИКОВ

Для удовлетворения потребностей той или иной отрасли производства необходимо обеспечить определенный график поступления сырья (или станков, или рабочей силы, или любых других производственных факторов). Обычно представляет затруднение согласование плана изменения спроса предприятий потребителей с возможностями поставщиков. Изменение спроса от этапа к этапу нарушает планомерную работу предприятий поставщиков. Если расходами на сокращение производства у поставщиков можно пренебречь, то увеличение очередного выпуска продукции требует дополнительных затрат. Расходы на увеличение производства естественно принять пропорциональными размеру увеличения выпуска продукции *). Инерционность поставщиков может в отдельных случаях приводить к поступлению избыточного количества требуемых производственных факторов. Это, в свою очередь, приводит к затратам. Содержание избыточных материалов, обслуживание и обеспечение излишнего персонала, до тех пор пока он не понадобится производству, связаны с бесполезными расходами.

*) В принятой ниже формальной схеме задачи предполагается, что расходы на расширение производства определяются только требуемым ростом продукции и не зависят от того, было ли до этого сокращение производства или нет.

Таким образом, с одной стороны, стремление точно удовлетворять спрос при изменяющихся во времени потребностях вызывает дорогостоящие реорганизации у поставщиков. С другой стороны, порядок снабжения, устраивающий поставщиков, может приводить к замораживанию материалов и оборудования. И тот и другой случай связаны с непроизводительными затратами. Возникает задача о составлении графика снабжения, при котором неизбежные дополнительные расходы сводились бы к минимуму.

К аналогичным вопросам приводит и ряд других ситуаций, связанных с согласованием взаимоотношений ряда организаций, которые, помимо общих интересов, имеют и внутренние частные задачи.

Приведем формальную постановку задачи [50] о составлении графика, обеспечивающего эффективный компромисс между противоречивыми требованиями к изучаемым процессам.

Пусть $a_1, a_2, \dots, a_n$ — потребности некоторой отрасли производства в определенном производственном факторе на все n периодов, на которые рассчитана программа работы. Будем для определенности понимать под производственным фактором, о котором идет речь, станки. Введем следующие обозначения: x_j — число станков, поставленных в j -й месяц; s_j — число станков, поступивших в первые j месяцев, но не использованных в течение этого времени. Подразумевается, что станки не используются только тогда, когда их наличие превышает потребности производства.

Расходы по содержанию избыточного количества станков пропорциональны величине

$$s_1 + s_2 + \dots + s_n.$$

Затраты на форсированное снабжение в отдельные месяцы, как мы условились, пропорциональны размерам требуемых приращений

$$(x_2 - x_1)_+ + (x_3 - x_2)_+ + \dots + (x_n - x_{n-1})_+.$$

Индекс $(+)$ означает, что соответствующее приращение учитывается только, если оно положительно:

$$(x_k - x_{k-1})_+ = \begin{cases} x_k - x_{k-1}, & \text{если } x_k - x_{k-1} > 0, \\ 0 & , \text{если } x_k - x_{k-1} \leq 0. \end{cases}$$

Суммарные дополнительные расходы, связанные с трудностями согласования возможностей поставщиков с требованиями потребителей, могут быть записаны в виде следующего выражения:

$$s_1 + s_2 + \dots + s_n + k [(x_2 - x_1)_+ + (x_3 - x_2)_+ + \dots + (x_n - x_{n-1})_+]. \quad (2.15)$$

Величина k определяет соотношение между расходами на форсирование поставок и ущербом, определяемым содержанием такого же количества излишнего оборудования. Если это соотношение в силу тех или иных причин изменяется со временем, то соответствующие коэффициенты следует ставить при каждом приращении.

Искомые переменные x_j и s_j связаны между собой следующей группой соотношений:

$$x_1 + x_2 + \dots + x_j = a_1 + a_2 + \dots + a_j + s_j \quad (2.16) \\ (j = 1, 2, \dots, n).$$

Каждое из этих равенств означает, что общее количество станков, поставленных за j месяцев, превышает потребности на s_j единиц.

В задаче требуется выбрать величины $x_j \geq 0$ и $s_j \geq 0$ таким образом, чтобы форма (2.15) достигла минимума при условиях (2.16). Эта задача пока еще не является задачей линейного программирования из-за особенностей формы (2.15). Однако ее можно легко свести к задаче линейного программирования. Точнее, можно указать такую задачу линейного программирования, вообще говоря, не эквивалентную данной, которая тем не менее имеет решение, реализующее оптимальное значение формы (2.15) при условиях (2.16).

Введем соотношения

$$\begin{aligned} x_2 - x_1 &= y_2 - z_2, \\ \cdot &\cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \\ \cdot &\cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \\ \cdot &\cdot \cdot \cdot \cdot \cdot \cdot \cdot \cdot \\ x_n - x_{n-1} &= y_n - z_n, \end{aligned}$$

где y_j и z_j неотрицательны ($y_j \geq 0$, $z_j \geq 0$), и рассмотрим следующую вспомогательную задачу линейного программирования.

Требуется обратить в минимум линейную форму

$$L = s_1 + s_2 + \dots + s_n + k(y_2 + y_3 + \dots + y_n)$$

при условиях

$$x_1 + x_2 + \dots + x_j - s_j = a_1 + a_2 + \dots + a_j$$

$$(j = 1, 2, \dots, n),$$

$$x_j - x_{j-1} = y_j - z_j \quad (j = 2, 3, \dots, n),$$

$$x_j \geq 0, \quad y_j \geq 0, \quad z_j \geq 0, \quad s_j \geq 0.$$

Нетрудно видеть, что решение этой задачи является также решением задачи (2.15) — (2.16). Действительно, для оптимального плана приведенной задачи линейного программирования при каждом j либо y_j , либо z_j равны нулю. Пусть это не так, и для некоторого $j = j_0$ решение задачи линейного программирования определяется величинами

$$y_{j_0} > 0 \text{ и } z_{j_0} > 0.$$

Тогда возможен один из двух случаев — либо $y_{j_0} > z_{j_0} > 0$, либо $z_{j_0} \geq y_{j_0} > 0$.

Первый случай не может соответствовать решению, так как при замене y_{j_0} на $\bar{y}_{j_0} = y_{j_0} - z_{j_0} > 0$ и z_{j_0} на $\bar{z}_{j_0} = 0$ условия задачи по-прежнему удовлетворяются, а линейная форма при этом уменьшается ($\bar{y}_{j_0} = y_{j_0} - z_{j_0} < y_{j_0}$).

Точно так же и второй случай не может соответствовать оптимальному плану. $\bar{y}_{j_0} = 0$ и $\bar{z}_{j_0} = z_{j_0} - y_{j_0} \geq 0$ удовлетворяют условиям, но уменьшают величину линейной формы, поскольку $\bar{y}_{j_0} = 0 < y_{j_0}$.

Таким образом, решение задачи линейного программирования приводит к значениям $y_{j_0} > 0$, когда $y_j - z_j = x_j - x_{j-1} > 0$, и к $y_j = 0$ при $y_j - z_j = x_j - x_{j-1} \leq 0$. Каждый план исходной задачи является планом вспомогательной задачи. С другой стороны, как мы видели, решение вспомогательной задачи является планом исходной задачи.

Следовательно, оптимальный план вспомогательной задачи линейного программирования является также решением задачи о составлении оптимального графика.

В связи с рассмотренной задачей уместно сделать следующее замечание.

Коэффициент k в формуле (2.15) характеризует соотношение дополнительных затрат поставщика и потребителя, отнесенных к единице производственного фактора. Далеко не во всех случаях планирующие организации располагают достаточным материалом для точного вычисления величины k . Чаше приходится вести расчет для ряда неотрицательных значений коэффициента k из некоторого заранее определенного диапазона. В таких случаях весьма полезным оказывается параметрическое линейное программирование, разработанное Гассом [46].

В приведенной ниже таблице выписаны компоненты векторов условий и вектора ограничений вспомогательной задачи линейного программирования, к которой сводится проблема составления оптимального графика.

	$A_1^{(x)}$	$A_j^{(x)}$	$A_j^{(y)}$	$A_j^{(z)}$	$A_j^{(s)}$	B
1	1	0	0	0	0	a_1
2	1	0	0	0	0	$a_1 + a_2$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$j-1$	1	0	0	0	0	$a_1 + \dots + a_{j-1}$
j	1	1	0	0	-1	$a_1 + \dots + a_j$
$j+1$	1	1	0	0	0	$a_1 + \dots + a_{j+1}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	1	1	0	0	0	$a_1 + \dots + a_n$
$n+1$	-1	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$

	$A_1^{(x)}$	$A_j^{(x)}$	$A_j^{(y)}$	$A_j^{(z)}$	$A_j^{(s)}$	B
$n + j - 2$	0	0	0	0	0	0
$n + j - 1$	0	1	-1	1	0	0
$n + j$	0	-1	0	0	0	0
$n + j + 1$	0	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2n - 1$	0	0	0	0	0	0

В столбцах $A_j^{(x)}$, $A_j^{(y)}$, $A_j^{(z)}$ $j = 2, 3, \dots, n$.

§ 8. ДИЛЕММА: БЫСТРО, НО ДОРОГО, ИЛИ МЕДЛЕННО, НО ДЕШЕВО

В ряде задач приходится делать выбор между дорогостоящими методами, позволяющими, однако, быстро достигнуть желанного результата, и более дешевыми методами, требующими большей затраты времени. «Стоимость» метода, естественно, в разных приложениях истолковывается по-своему. В одних случаях «стоимость» измеряется затратами денег или материалов. В других случаях «стоимость» метода определяют факторы, которые являются лимитирующими только в условиях данной задачи.

Покажем на примере производственной задачи, как можно сводить проблему выбора между дорогостоящим, но быстрым, и дешевым, но медленным, к задаче линейного программирования [49].

Пусть программа работы предприятия по определенному заказу рассчитана на n этапов (дней, месяцев). Пусть далее в соответствии с графиком на j -м этапе требуется r_j штук специальных инструментов ($j = 1, 2, \dots, n$). К концу этапа инструмент изнашивается и требует ремонта. Нормальный ремонт обходится предприятию в b рублей за один инструмент. Время, необходимое на нормальный ремонт, равно длительности p этапов

работы. Срочный ремонт обходится предприятию дороже — c рублей ($c > b$) за единицу. Однако при этом инструмент возвращается быстрее на производство и через q этапов ($q < p$) он уже может быть использован в работе предприятия. Приобретение нового инструмента обходится предприятию в a рублей за единицу. Естественно предполагать, что $a > c$. В противном случае было бы нерационально ремонтировать инструмент.

Возникает задача, как должно предприятие планировать ремонт и приобретение оборудования, чтобы при минимальных издержках удовлетворять свои потребности в инструменте в продолжение всего периода выполнения заказа.

Переведем задачу на формальный математический язык.

Пусть x_j количество инструмента, приобретенного для работы на j -м этапе. Остальные потребности, если они имеются, удовлетворяются отремонтированным инструментом.

Обозначим через y_j и z_j число инструментов, отправленных к концу j -го этапа соответственно в нормальный и срочный ремонт. Не всегда имеет смысл немедленно отправлять в ремонт весь изношенный инструмент. Пусть s_j — запас использованного инструмента на j -м этапе, зафиксированный после того, как $y_j + z_j$ инструментов направлено в ремонт. Этот запас весь или частично может быть отправлен в ремонт на следующем этапе вместе с частью или со всеми инструментами, использованными на $(j + 1)$ -м этапе.

Ясно, что количество инструмента, использованного на j -м этапе (r_j), складывается из числа инструментов, отправленных в ремонт к концу этого этапа ($y_j + z_j$), и изменения (по сравнению с предыдущим этапом) в запасе изношенного инструмента $s_j - s_{j-1}$, т. е.

$$y_j + z_j + s_j - s_{j-1} = r_j \quad (j = 1, 2, \dots, n); \quad s_0 = 0. \quad (2.17)$$

Число инструментов, возвращенных на j -м этапе из нормального ремонта, равно числу инструментов (y_{j-p}), направленных в ремонт за p этапов до этого. Точно так же число инструментов, возвращенных на j -м этапе из срочного ремонта, совпадает с количеством инструмента (z_{j-q}), направленным в ремонт за q этапов до этого.

Потребности в инструменте на j -м этапе (r_j) покрываются за счет приобретения нового инструмента (x_j) и инструмента, возвращаемого из нормального (y_{j-p}) и срочного (z_{j-q}) ремонта, т. е.

$$\left. \begin{aligned} x_j &= r_j \quad (j=1, 2, \dots, q), \\ x_j + z_{j-q} &= r_j \quad (j=q+1, q+2, \dots, p) \\ x_j + y_{j-p} + z_{j-q} &= r_j \quad (j=p+1, p+2, \dots, n). \end{aligned} \right\} \quad (2.18)$$

Расходы предприятия на j -м этапе складываются из затрат на приобретение инструмента (ax_j), на нормальный ремонт (by_j) и на срочный ремонт (cz_j).

Общие издержки предприятия на приобретение и ремонт инструмента, обеспечивающего его потребности на все n этапов работы, равны

$$\sum_{j=1}^n (ax_j + by_j + cz_j). \quad (2.19)$$

Здесь по условию $a > c$, $c > b$.

Здравый смысл подсказывает, что за p этапов до завершения программы, для которой требуется специальный инструмент, следует прекратить нормальный ремонт, а за q этапов — срочный ремонт. Дело в том, что инструмент, отправленный в ремонт после указанного срока, все равно не вернется на предприятие до завершения программы. Таким образом, для обращения в минимум издержек предприятия необходимо, чтобы

$$\begin{aligned} y_{n-p+1} &= y_{n-p+2} = \dots = y_n = 0, \\ z_{n-q+1} &= z_{n-q+2} = \dots = z_n = 0. \end{aligned}$$

Следовательно, общие затраты на оборудование могут быть представлены в виде следующей линейной формы:

$$L = a \sum_{j=1}^n x_j + b \sum_{j=1}^{n-p} y_j + c \sum_{j=1}^{n-q} z_j. \quad (2.20)$$

Мы пришли к задаче линейного программирования. Требуется обратить в минимум линейную форму (2.20) при линейных условиях (2.17) и (2.18). Искомые переменные здесь величины x_j , y_j , z_j и s_j ($j = 1, 2, \dots, n$).

Все они означают число инструментов и поэтому неотрицательны

$$x_j \geq 0, y_j \geq 0, z_j \geq 0, s_j \geq 0 \quad (j=1, 2, \dots, n).$$

Собирая все условия, выпишем задачу в виде следующей системы

$$L = a \sum_{j=1}^n x_j + b \sum_{j=1}^{n-p} y_j + c \sum_{j=1}^{n-q} z_j,$$

$$x_j + z_{j-q} = r_j \quad (j=q+1, q+2, \dots, p),$$

$$x_j + y_{j-p} + z_{j-q} = r_j \quad (j=p+1, p+2, \dots, n),$$

$$y_j + z_j + s_j - s_{j-1} = r_j \quad (j=1, 2, \dots, n),$$

$$x_j \geq 0, y_j \geq 0, z_j \geq 0, s_j \geq 0 \quad (j=1, 2, \dots, n).$$

Учитывая, что $x_j = r_j$ при $j=1, 2, \dots, q$; $y_j = 0$ при $j > n-p$ и $z_j = 0$ при $j > n-q$, приходим к задаче линейного программирования с $4n - (r+p+q)$ переменными и $2n - q$ условиями.

Относительно простая форма записи векторов условий позволяет в частном случае, когда $p - q = 1$ (т. е. когда длительность нормального и срочного ремонта отличаются на время одного этапа работы) получить решение задачи в явном виде [49]. Рамки настоящей книги не позволяют рассмотреть этот вопрос.

§ 9. ВЫБОР РАЦИОНАЛЬНОЙ СИСТЕМЫ ДОПУСКОВ

В литературе по машиностроению и приборостроению много внимания уделяется вопросу о рациональной системе допусков на отдельные размеры деталей механизма. Система допусков считается рациональной, если она обеспечивает требуемую точность при наименьшей общей стоимости механизма.

Переведем на формальный язык задачу о выборе системы допусков. Будем при этом пользоваться терминологией и обозначениями, принятыми в теории точности механизмов.

Пусть в механизме имеются только скалярные ошибки. Тогда предельная ошибка механизма определяется по формуле

$$\xi^2 = a_1^2 \delta_1^2 + a_2^2 \delta_2^2 + \dots + a_n^2 \delta_n^2.$$

Г. Г. Баранова [1] зависимость стоимости и времени обработки от допуска предполагается более сложной и задается соотношением вида

$$R = A + \frac{B}{\delta^p}.$$

Заметим, что функции, которыми аппроксимируется зависимость стоимости от допуска, являются выпуклыми функциями. Хорда, соединяющая любые две точки графика зависимости $R = \varphi(\delta)$, проходит выше (точнее, не ниже) дуги кривой, соединяющей эти точки. Это и ясно: чем выше класс точности устройства, тем дороже обходится дальнейшее сокращение допусков. Больше того, как правило, можно считать, что стоимость представляет собой выпуклую функцию от квадрата допуска

$$R = \varphi(\delta^2).$$

В этих предположениях можно использовать методы линейного программирования для определения оптимальной системы допусков.

Стоимость изготовления всех деталей механизма связана с допусками на отдельные размеры деталей соотношением

$$R = \varphi_1(\delta_1^2) + \dots + \varphi_n(\delta_n^2),$$

где $\varphi_i(\delta_i^2)$ — выпуклая функция от δ_i^2 .

Введем обозначения

$$\delta_i^2 = \Delta_i, \quad a_{ik}^2 = \alpha_{ik}, \quad \xi_k^2 = \eta_k$$

$$(i = 1, 2, \dots, n; k = 1, 2, \dots, m).$$

Выражение для стоимости изготовления всех деталей принимает вид

$$R = \varphi_1(\Delta_1) + \dots + \varphi_n(\Delta_n), \quad (2.21)$$

а условия, обеспечивающие требуемую точность механизма во всем диапазоне его работы, представляют систему неравенств

[illegible]

Обычно диапазон возможных значений допусков для каждой детали механизма задается заранее. По-

этому величины Δ_i должны удовлетворять дополнительной системе условий

$$\Delta_{i \min} \leq \Delta_i \leq \Delta_{i \max}. \quad (2.23)$$

Если эти ограничения не заданы, то выбор Δ_i подчиняется лишь естественному дополнительному условию

$$\Delta_i \geq 0 \quad (i=1, 2, \dots, n).$$

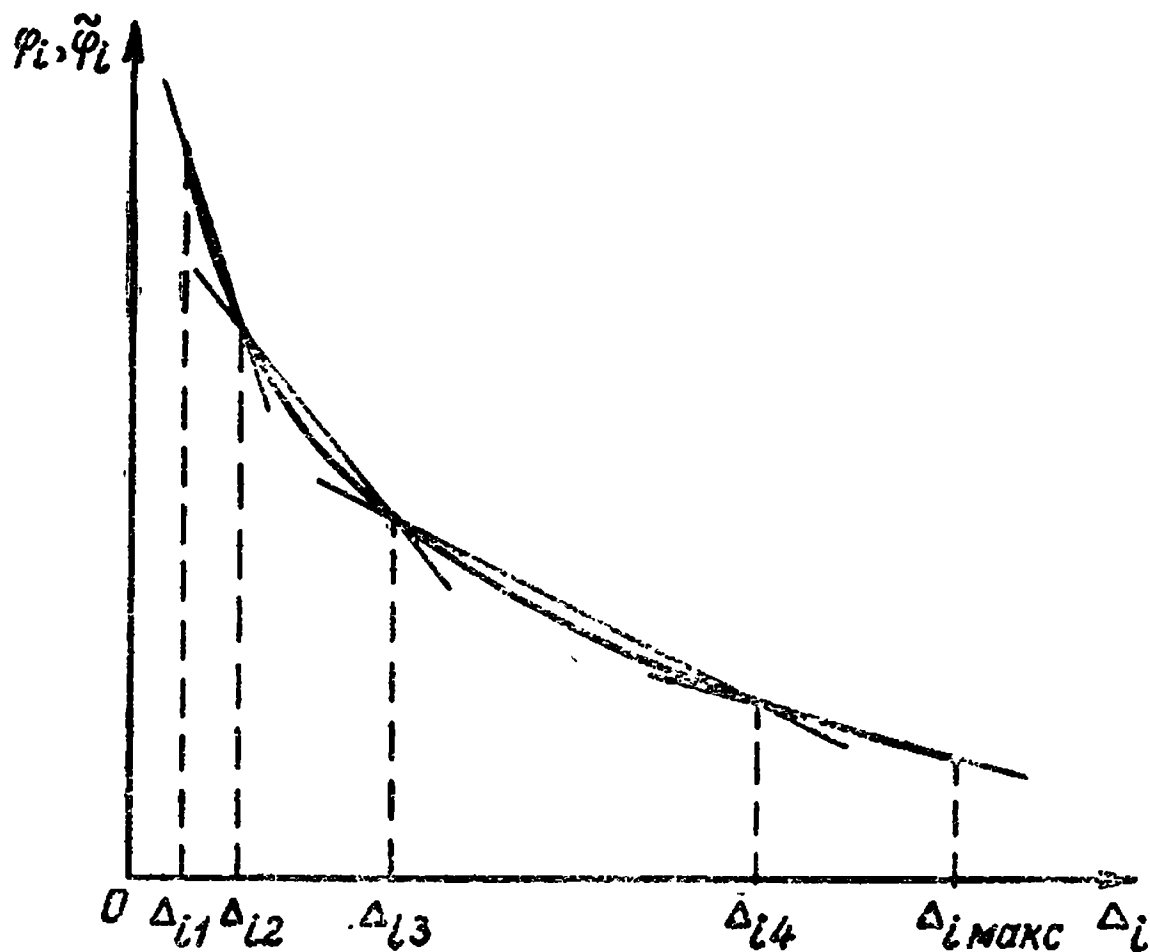


Рис. 2.1

Таким образом, задача об отыскании рациональной системы допусков сводится к определению набора неотрицательных чисел $\Delta_1, \Delta_2, \dots, \Delta_n$, обращающих в минимум форму R (2.21) при условиях (2.22) и (2.23). Это еще не задача линейного программирования. Однако, как мы увидим, рассматриваемая здесь задача может со сколь угодно высокой степенью точности решаться методами линейного программирования. $\varphi_i(\Delta_i)$, как мы условились, — выпуклая функция квадрата допуска $\delta_i^2 = \Delta_i$. Примерный график изменения φ_i в зависимости от Δ_i изображен на рис. 2.1.

Пусть диапазон изменения значений Δ_i равен $(\Delta_{i \min}, \Delta_{i \max})$, а наибольшая допустимая ошибка в вычислении оптимальной стоимости равна ϵ . Разделим диапазон изменения стоимости i -й детали $\varphi_i(\Delta_{i \max}) - \varphi_i(\Delta_{i \min})$ на

s_i частей так, чтобы хорды, соединяющие соседние точки деления, отстояли от кривой $\varphi_i(\Delta)$ (по оси ординат) не более чем на $\frac{\varepsilon}{n}$. Обозначим абсциссы точек деления $\varphi_i(\Delta_i)$ через Δ_{ij} .

Будем наряду с выпуклой функцией $\varphi_i(\Delta_i)$ рассматривать ломаную $\tilde{\varphi}_i(\Delta_i)$, совпадающую с $\varphi_i(\Delta_i)$ в точках Δ_{ij} .

Уравнения прямых, которым принадлежат звенья ломаной, записываются в виде

$$y_i = \lambda_{ij}\Delta_i + \mu_{ij},$$

где

$$\left. \begin{aligned} \lambda_{ij} &= \frac{\varphi_i(\Delta_{ij}) - \varphi_i(\Delta_{i,j-1})}{\Delta_{ij} - \Delta_{i,j-1}}, \\ \mu_{ij} &= \frac{\Delta_{ij}\varphi_i(\Delta_{i,j-1}) - \Delta_{i,j-1}\varphi_i(\Delta_{ij})}{\Delta_{ij} - \Delta_{i,j-1}} \end{aligned} \right\} \quad (2.24)$$

$$(i = 1, 2, \dots, n, j = 1, 2, \dots, s_i).$$

Обозначим

$$\tilde{R} = \tilde{\varphi}_1(\Delta_1) + \tilde{\varphi}_2(\Delta_2) + \dots + \tilde{\varphi}_n(\Delta_n).$$

Покажем, что $\min R$ при условиях (2.22), (2.23) отличается не больше чем на ε от минимума $\tilde{R}$ при тех же условиях.

Ясно, что $\min \tilde{R}$ и $\min R$, определенные на дискретном множестве точек Δ_{ij} , совпадают, поскольку эти точки являются общими для $\varphi_i(\Delta_i)$ и $\tilde{\varphi}_i(\Delta_i)$.

Если же не ограничивать область изменения дискретным множеством Δ_{ij} , то минимумы R и $\tilde{R}$, вообще говоря, достигаются в разных точках и не совпадают между собой. Однако, как легко видеть, различие между ними не превысит ε . Действительно, из самого построения $\tilde{\varphi}_i(\Delta_i)$ [$\varphi_i(\Delta)$ — выпуклая функция] следует, что

$$\tilde{\varphi}_i(\Delta_i) \geq \varphi_i(\Delta_i) \quad (2.25)$$

и

$$\tilde{\varphi}_i(\Delta_i) - \varphi_i(\Delta_i) \leq \frac{\varepsilon}{n}. \quad (2.26)$$

Суммируя неравенства по i от 1 до n и учитывая введенные обозначения, получим

$$\tilde{R}(\Delta) \geq R(\Delta)$$

и

$$\tilde{R}(\Delta) - R(\Delta) \leq \varepsilon,$$

где

$$\Delta = \{\Delta_1, \Delta_2, \dots, \Delta_n\}.$$

Пусть $R(\Delta)$ достигает при условиях (2.22) и (2.23) минимума в точке $\Delta^\circ = \{\Delta_1^\circ, \dots, \Delta_n^\circ\}$. Тогда $\tilde{R}(\Delta^\circ) \geq R(\Delta^\circ) = \min R$ и

$$\tilde{R}(\Delta^\circ) - R(\Delta^\circ) \leq \varepsilon. \quad (2.27)$$

Пусть далее $\tilde{R}(\Delta') = \min \tilde{R}$ при условиях (2.22), (2.23). Здесь

$$\Delta' = \{\Delta_1', \dots, \Delta_n'\}.$$

В соответствии с построением $R(\Delta)$ и $\tilde{R}(\Delta)$ и определением Δ° и Δ' имеем

$$\tilde{R}(\Delta^\circ) \geq \tilde{R}(\Delta') \geq R(\Delta') \quad (2.28)$$

и

$$R(\Delta^\circ) \leq R(\Delta'). \quad (2.29)$$

Сопоставляя неравенства (2.27) и (2.28), получаем

$$R(\Delta^\circ) \geq R(\Delta') - \varepsilon. \quad (2.30)$$

Неравенства (2.29) и (2.30) означают, что

$$|R(\Delta') - R(\Delta^\circ)| \leq \varepsilon. \quad (2.31)$$

Следовательно, задача определения $\min R$ при условиях (2.22), (2.23) может быть с любой заранее заданной точностью заменена задачей вычисления $\min \tilde{R}$ при тех же условиях. Последняя же задача, как мы сейчас увидим, может быть решена методами линейного программирования.

Рассмотрим следующую задачу.

Требуется найти минимум линейной формы

$$L = \sum_{i=1}^n y_i$$

при условиях

$$y_i \geq \tilde{\varphi}_i(\Delta_i),$$

$$\sum_{i=1}^n \alpha_{ik} \Delta_i \leq \eta_k,$$

$$\Delta_{i \min} \leq \Delta_i \leq \Delta_{i \max},$$

$$(i = 1, 2, \dots, n; k = 1, 2, \dots, m).$$

Очевидно, задача о минимизации $\tilde{R}$ может быть записана в аналогичной форме. Однако область изменения переменных здесь уже. Эта задача может быть сформулирована следующим образом.

Требуется вычислить минимум

$$\tilde{R} = \sum_{i=1}^n y_i$$

при условиях

$$y_i = \tilde{\varphi}_i(\Delta_i),$$

$$\sum_{i=1}^n \alpha_{ik} \Delta_i \leq \eta_k,$$

$$\Delta_{i \min} \leq \Delta_i \leq \Delta_{i \max}$$

$$(i = 1, 2, \dots, n; k = 1, 2, \dots, m).$$

Вместо условий $y_i \geq \tilde{\varphi}_i(\Delta_i)$, при которых требуется вычислить $\min L$, при определении $\min \tilde{R}$ необходимо учесть более жесткие требования $y_i = \tilde{\varphi}_i(\Delta_i)$.

Для выпуклых функций условие

$$y_i \geq \tilde{\varphi}_i(\Delta_i) \tag{2.32}$$

может быть записано в виде

$$y_i \geq \lambda_{ij} \Delta_i + \mu_{ij}, j = 1, 2, \dots, s_i, \tag{2.33}$$

где λ_{ij} и μ_{ij} определяются по формулам (2.24). Эквивалентность условий (2.32) и (2.33) — очевидное свой-

ство выпуклых функций. Область, расположенная над ломаной, вписанной в выпуклую кривую, располагается также выше каждой прямой, содержащей отрезок -- звено ломаной. Отметим, что такая замена нелинейных условий линейными характерна только для выпуклой функции $\varphi_i(\Delta_i)$. *) Следует также подчеркнуть, что неравенства (2.33) имеют место для всей области изменения Δ_i от $\Delta_{i \min}$ до $\Delta_{i \max}$.

Таким образом, мы пришли к следующей задаче линейного программирования.

Требуется определить минимум линейной формы

$$L = \sum_{i=1}^n y_i$$

при линейных условиях

$$y_i \geq \lambda_{ij} \Delta_i + \mu_{ij}, \quad (2.33)$$

$$\sum_{i=1}^n \alpha_{ik} \Delta_i \leq \eta_k, \quad (2.22)$$

$$\Delta_{i \min} \leq \Delta_i \leq \Delta_{i \max} \quad (2.23)$$

$$(i = 1, 2, \dots, n; k = 1, 2, \dots, m).$$

Легко видеть, что L достигнет минимума, когда неравенства (2.32) обратятся в равенства. Действительно, это условие не скажется на ограничениях (2.22) и (2.23), а линейная форма с уменьшением y_i уменьшится. Это значит, что $\min L$ совпадает с $\min \tilde{R}$ и, следовательно, приближенное вычисление целесообразной системы допусков сводится к решению задачи линейного программирования.

§ 10. ПЛАНИРОВАНИЕ ПРОИЗВОДСТВА И ПЕРЕВОЗОК

Рассмотрим несколько типичных задач, возникающих в работе планирующих организаций на различных уровнях. Начнем с простейшей задачи о перевозке однородного продукта и будем постепенно вводить усложнения,

*) К аналогичным результатам приходим при решении задач на максимум формы $R = \sum_{i=1}^n \varphi_i(\Delta_i)$ при линейных условиях, когда φ_i -- вогнутые функции Δ_i .

расширяющие область применения линейного программирования к планированию экономики.

Пусть имеется m пунктов производства продукта и n пунктов потребления. Производство и потребление сбалансированы, т. е. общий объем продукции, производимой в m пунктах производства, полностью покрывает спрос и потребляется в n пунктах потребления. Задача заключается в определении рациональной системы перевозок, при которой общие транспортные расходы будут минимальны. При этом должны быть учтены следующие условия:

1°. Спрос каждого пункта потребления полностью удовлетворяется.

2°. Весь произведенный в каждом пункте производства продукт полностью вывозится в пункты потребления.

3°. Пропускная способность транспорта ограничена. Введем следующие обозначения:

x_{ij} — количество единиц продукта, перевозимого из i -го пункта производства в j -й пункт потребления;

c_{ij} — стоимость перевозки единицы продукта из i -го пункта в j -й;

d_{ij} — ограничение пропускной способности транспорта на линии (i, j) (в единицах продукта);

a_i — количество единиц продукта, производимое в i -м пункте;

b_j — количество единиц продукта, потребляемое в j -м пункте.

Все параметры относятся к определенному отрезку времени, для которого составляется график перевозок.

В принятых обозначениях задача сводится к обращению в минимум линейной формы

$$L = L(x_{ij}) = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij}$$

при условиях

$$1^\circ. \sum_{i=1}^m x_{ij} = b_j \text{ — спрос в } j\text{-м пункте } (j = 1, 2, \dots, n).$$

$$2^\circ. \sum_{j=1}^n x_{ij} = a_i \text{ — производство в } i\text{-м пункте } (i = 1, 2, \dots, m).$$

3°. $0 \leq x_{ij} \leq d_{ij}$ — ограничения по пропускной способности ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$).

По условию

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j.$$

Полученную задачу принято называть транспортной задачей с ограничениями по пропускным способностям.

Из перечисленных условий трудоемкость вычислений определяют, главным образом, ограничения вида 1° и 2°. Методы линейного программирования, изложенные в главе 5, могут быть несколько скорректированы и обобщены таким образом, чтобы ограничения типа 3° не вносили заметных усложнений в расчеты. Однако ограниченные рамки настоящего руководства вынуждают нас опустить изложение этого вопроса.

Усложним теперь задачу.

Пусть в j -м пункте спрос возрос на β_j . Чтобы удовлетворить спрос, необходимо увеличить производство в некоторых пунктах и ввести дополнительные перевозки. Требуется разработать задания пунктам производства и план дополнительных перевозок, чтобы, учитывая ограниченные возможности расширения производства в каждом пункте и ограниченную пропускную способность транспорта, удовлетворить спрос при минимальных расходах.

Переведем задачу на формальный язык. Введем следующие обозначения.

p_i — затраты на дополнительное производство единицы продукта в i -м пункте *);

u_i — величина дополнительного производства в i -м пункте;

f_i — ограничение по возможности расширения производства в i -м пункте;

*) Здесь и в дальнейшем под затратами на дополнительное производство и расширение пропускной способности транспорта подразумеваются приведенные затраты, при расчете которых учтены коэффициенты, позволяющие сравнивать и суммировать затраты (текущие и капиталовложения) в сопоставимых единицах.

y_{ij} — дополнительное количество единиц продукта, перевозимое из i -го пункта в j -й;

$\tilde{d}_{ij}$ — ограничение пропускной способности транспорта по линии (i, j) , учитывающее график основных перевозок ($\tilde{d}_{ij} = d_{ij} - x_{ij}$, где x_{ij} — перевозка в направлении (i, j) , определенная в результате решения предыдущей задачи);

c_{ij} , a_i , b_j , d_{ij} имеют тот же смысл, что и в предыдущей задаче.

Во введенных обозначениях задача формулируется следующим образом.

Требуется обратить в минимум линейную форму

$$L = L(y_i, y_{ij}) = \sum_{i=1}^m p_i y_i + \sum_{i=1}^m \sum_{j=1}^n c_{ij} y_{ij}$$

при условиях:

$$\sum_{i=1}^m y_{ij} = \beta_j \text{ — дополнительный спрос в } j\text{-м пункте} \\ (j = 1, 2, \dots, n);$$

$$\sum_{j=1}^n y_{ij} = y_i \text{ — дополнительное производство в } i\text{-м} \\ \text{пункте } (i = 1, 2, \dots, m);$$

$$y_i \leq f_i \text{ — ограничения по возможностям расши-} \\ \text{рения производства в } i\text{-м пункте} \\ (i = 1, 2, \dots, m);$$

$$0 \leq y_{ij} \leq \tilde{d}_{ij} \text{ — ограничения по транспорту.}$$

Нетрудно видеть, что эта задача имеет ту же структуру, что и предыдущая. Действительно, исключая переменные y_i , получаем следующую задачу. Требуется обратить в минимум линейную форму

$$L = \sum_{i=1}^m \sum_{j=1}^n (c_{ij} + p_i) y_{ij}$$

при условиях

$$\sum_{i=1}^m y_{ij} = \beta_j, \quad \sum_{j=1}^n y_{ij} \leq f_i, \quad 0 \leq y_{ij} \leq \tilde{d}_{ij}.$$

Здесь естественно полагать

$$\sum_{j=1}^n \beta_j \leq \sum_{i=1}^m f_i.$$

Легко видеть, что введение фиктивного пункта спроса с объемом потребления, равным

$$\beta_{n+1} = \sum_{i=1}^m f_i - \sum_{j=1}^n \beta_j,$$

позволяет свести эту задачу к предыдущей. Величина β_{n+1} определяет неиспользованные возможности по расширению производства.

Обобщая задачу, мы предполагали, что увеличение спроса, расширение производства и введение дополнительных перевозок не будут нарушать установившихся перевозок (x_{ij}) .

Вообще говоря, это не так. Если дополнительные перевозки рассчитаны на длительный срок, изменение графика перевозок может иметь существенный экономический эффект. Задача в этом случае будет отличаться от предыдущей тем, что в качестве переменных придется рассматривать не дополнительные перевозки y_{ij} , а суммарные перевозки z_{ij} из i -го пункта производства в j -й пункт потребления. Соответственно изменятся и условия задачи.

При сохранении ранее введенных обозначений формальная структура задачи может быть представлена следующим образом.

Требуется обратить в минимум линейную форму

$$L = L(y_i, z_{ij}) = \sum_{i=1}^m p_i y_i + \sum_{j=1}^n \sum_{i=1}^m c_{ij} z_{ij}$$

при условиях

$$\sum_{i=1}^m z_{ij} = b_j + \beta_j \text{ — спрос в } j\text{-м пункте } (j = 1, 2, \dots, n),$$

$$\sum_{j=1}^n z_{ij} \leq a_i + y_i \text{ — производство в } i\text{-м пункте } (i = 1, 2, \dots, m),$$

$$0 \leq z_{ij} \leq d_{ij} \text{ — ограничения по пропускной способности транспорта,}$$

$$0 \leq y_i \leq f_i \text{ — ограничения по возможностям расширения производства в } i\text{-м пункте } (i = 1, 2, \dots, m).$$

В ряде случаев естественно потребовать, чтобы по новому плану перевозок из каждого пункта производства вывозилось не менее, чем по старому плану (до расширения производства).

$$\sum_{j=1}^n z_{ij} \geq \sum_{j=1}^n x_{ij} = a_i. \quad (*)$$

Тогда эта задача так же, как и предыдущая, сводится к транспортной проблеме с ограничениями по пропускным способностям.

К естественному усложнению задачи приводят следующие рассуждения.

Сравнивая решения рассмотренных в настоящем параграфе задач, определяем

$$\Delta = L(y_i, z_{ij}) - L(x_{ij})$$

— стоимость удовлетворения возросшего спроса при рациональном планировании производства и перевозок. Может оказаться, что суммарные расходы уменьшатся, если часть средств будет направлена на расширение пропускной способности транспорта.

Следующая, более широкая постановка задачи планирования указывает целесообразный подход к распределению капиталовложений.

Пусть параметр w_{ij} характеризует увеличение пропускной способности транспорта, обеспечивающего перевозки из i -го пункта в j -й, а q_{ij} определяет затраты на увеличение пропускной способности линии (i, j) на единицу перевозимого продукта *).

Сохраняя предыдущие обозначения, можно сформулировать следующую задачу.

Требуется минимизировать линейную форму $n + 2mn$ переменных (y_i, z_{ij}, w_{ij})

$$L = L(y_i, z_{ij}, w_{ij}) = \sum_{i=1}^m p_i y_i + \sum_{j=1}^n \sum_{i=1}^m c_{ij} z_{ij} + \sum_{j=1}^n \sum_{i=1}^m q_{ij} w_{ij}$$

при условиях

$$\sum_{i=1}^m z_{ij} = b_j + \beta_j,$$

* См. сноску на стр. 112.

$$\sum_{j=1}^n z_{ij} \leq a_i + y_i,$$

$$0 \leq z_{ij} \leq d_{ij} + w_{ij},$$

$$0 \leq y_i \leq f_i,$$

$$w_{ij} \geq 0 \quad (i=1, 2, \dots, m; j=1, 2, \dots, n).$$

Здесь удобно рассматривать два случая в зависимости от того, выполняются ли условия (*) или нет.

Пусть вначале требования (*) приняты.

Тогда, исключая переменные y_i и вводя дополнительные переменные $z_{i, n+1}$, приходим к следующей записи задачи.

Требуется обратить в минимум линейную форму

$$L = \sum_{i=1}^m \sum_{j=1}^n (p_i + c_{ij}) z_{ij} + \sum_{i=1}^m \sum_{j=1}^n q_{ij} w_{ij} + \text{const}$$

при условиях

$$\sum_{i=1}^m z_{ij} = \begin{cases} b_j + \beta_j & \text{при } j=1, 2, \dots, n; \\ \sum_{k=1}^m f_k - \sum_{l=1}^n \beta_l & \text{при } j=n+1; \end{cases}$$

$$\sum_{j=1}^{n+1} z_{ij} = a_i + f_i, \quad i=1, 2, \dots, m;$$

$$0 \leq z_{ij} \leq \begin{cases} d_{ij} + w_{ij} & \text{при } j=1, 2, \dots, n; i=1, 2, \dots, m; \\ f_i & \text{при } j=n+1, i=1, 2, \dots, m; \end{cases}$$

$$w_{ij} \geq 0, \quad i=1, 2, \dots, m; j=1, 2, \dots, n.$$

Если система чисел $\{z_{ij}, w_{ij}\}$ является решением этой задачи, то оптимальный план производства и перевозок определяется набором чисел $\{y_i, z_{ij}, w_{ij}\}$, где $y_i = f_i - z_{i, n+1}$.

Условия (*) не всегда приемлемы.

В частности, если имеется возможность существенного расширения производства в районах, близких к потребителю, может оказаться целесообразным сократить перевозки из дальних пунктов производства по сравне-

нию с запланированными ранее (до увеличения спроса и увеличения производства).

Не будем теперь требовать выполнения условий (*). Тогда, исключая переменные y_i и вводя дополнительные переменные v_i и $z_{i,n+1}$, можно придать задаче планирования производства и перевозок однородного продукта следующий вид.

Требуется обратить в минимум линейную форму

$$L = \sum_{i=1}^m \sum_{j=1}^n c_{ij} z_{ij} + \sum_{i=1}^m p_i (v_i - z_{i,n+1}) + \sum_{i=1}^m \sum_{j=1}^n q_{ij} w_{ij} + \text{const}$$

при условиях

$$\sum_{i=1}^m z_{ij} = \begin{cases} b_j + \beta_j & \text{при } j = 1, 2, \dots, n; \\ \sum_{k=1}^m f_k - \sum_{l=1}^n \beta_l & \text{при } j = n + 1; \end{cases}$$

$$\sum_{j=1}^{n+1} z_{ij} = a_i + f_i, \quad i = 1, 2, \dots, m;$$

$$0 \leq z_{ij} \leq \begin{cases} d_{ij} + w_{ij} & \text{при } j = 1, 2, \dots, n; i = 1, 2, \dots, m; \\ f_i + v_i & \text{при } j = n + 1, i = 1, 2, \dots, m; \end{cases}$$

$$v_i \geq 0; w_{ij} \geq 0, \quad i = 1, 2, \dots, m; j = 1, 2, \dots, n.$$

Переменные v_i определяют возможные сокращения вывоза из i -го пункта. Если сокращение вывоза из i -го пункта связано с дополнительными расходами, то коэффициент при v_i в линейной форме должен быть соответствующим образом увеличен.

Если система чисел $\{z_{ij}, v_i, w_{ij}\}$ является решением последней задачи, то оптимальный план производства и перевозок, определяется набором чисел $\{z_{ij}, y_i, w_{ij}\}$, где

$$y_i = \begin{cases} f_i - z_{i,n+1} & \text{при } z_{i,n+1} \leq f_i, \quad i = 1, 2, \dots, m; \\ 0 & \text{при } z_{i,n+1} > f_i. \end{cases}$$

При этом предполагается, что компоненты плана не подчинены требованиям (*).

Теперь целесообразно усложнить задачу и ввести в рассмотрение систему неоднородных продуктов. Здесь появляются новые линейные ограничения, характеризующие возможную взаимозаменяемость отдельных продуктов и минимальную потребность разных пунктов в каждой категории продукции. Кроме того, при постановке задачи следует учесть, что перевозка разнородных продуктов, как правило, не может быть обеспечена однородным транспортом.

Усложнение задач в указанных направлениях не приводит к новым принципиальным трудностям. Но с увеличением числа пунктов производства и потребления или количества категорий продукции и транспортных связей число переменных и ограничений быстро растет. Еще быстрее растет трудоемкость вычислений в соответствующих задачах линейного программирования. Относительно ограниченное быстродействие современных вычислительных машин (несколько десятков тысяч операций в секунду) и, главным образом, их ограниченная память не позволяют пока ставить и решать задачи планирования народного хозяйства страны в полном объеме. Заметим, что в настоящее время имеются трудности в сборе и систематизации достаточных исходных данных для постановки широких задач детального планирования экономики.

Представляется целесообразным на первых этапах приложения методов линейного программирования к планированию экономики не стремиться к детальному рассмотрению широких задач. Вначале в качестве отдельных пунктов следует рассматривать обширные районы и варьировать только теми переменными, которые определяют узкие места экономики. Остальные параметры закрепляют теми их значениями, которые сложились в результате многолетней практики планирования. Так, постепенно наращивая сложность задачи, вводя в рассмотрение детали последовательно по мере их важности, можно шаг за шагом отрабатывать рациональные планы производства, потребления и перевозок. Это сложная задача. Однако принципиальных трудностей в ее решении нет. Экономический эффект, который следует ожидать от научного планирования, трудно переоценить, если учесть, что только сжегодные транспортные расходы в СССР составляют 100 миллиардов рублей. Плановое

хозяйство нашей страны открывает широкий простор применению линейного программирования к самым разнообразным задачам народнохозяйственного планирования.

§ 11. ВОЕННЫЕ ПРИЛОЖЕНИЯ МЕТОДОВ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ (ПО МАТЕРИАЛАМ ЗАРУБЕЖНОЙ ПЕЧАТИ)

Решение военных вопросов в большей степени, чем анализ любых других проблем, требует четкого планирования и тщательного учета ограничений. Это относится как к решению тактических задач, так и к военно-хозяйственным вопросам и особенно к проектированию аппаратуры для управления военной техникой и боевыми действиями. Поэтому неудивительно, что простые и легко переводимые на язык вычислительных машин методы линейного программирования нашли широкое применение в разнообразных военных задачах. За последние годы в иностранной печати появилось большое количество статей, так или иначе связанных с применением схем линейного программирования к анализу методов снабжения, порядка ремонта оборудования, к целераспределению, к выбору рациональной системы вооружения и к решению игровых задач, возникающих в самых разнообразных военных проблемах.

Следует подчеркнуть некоторое различие народнохозяйственных и военных приложений линейного программирования. Постановка задач, относящихся к планированию экономики, обычно вызывает меньше принципиальных затруднений, чем постановка военных задач. Понятие «показателя качества — критерия выгоды» в хозяйственных проблемах является более определенным, чем в военных задачах. Условия, позволяющие выделить замкнутую задачу планирования и указать ее связь со смежными вопросами, легче обозримы и проще проверяются для экономических задач, чем для военных приложений. Достаточно сослаться хотя бы на то, что в народнохозяйственных задачах сравнительно короткий срок отделяет постановку задачи от ее всесторонней практической проверки по реализации решения. В военных задачах определение соответствия математической модели практическим потребностям может про-

изойти тогда, когда уже не будет возможности ввести необходимые коррективы. Военные задачи чаще, чем народнохозяйственные вопросы, приходится решать в условиях неполной информации — в условиях неопределенности. Тем не менее, применение линейного программирования к решению военных задач управления и планирования не только полезно, но и необходимо.

При решении задач управления военными комплексами и системами и военном планировании имеют дело с огромными потоками информации, неполностью и недостаточно четко передаваемой участникам управления и планирования, и некоторой дезинформацией. Только жесткие рамки математической модели могут фильтровать этот поток, выделяя существенное, исследуя неполное и исключая непоследовательное. Модели линейного программирования в некоторых случаях могут автоматически справляться с ситуациями, которые обычно требуют суждений и опыта многих людей. Они способны суммировать большое количество информации и успешно координировать действия штабов различного уровня.

В последующих параграфах мы рассмотрим несколько примеров, которые характеризуют, но отнюдь не исчерпывают возможности военных приложений линейного программирования.

§ 12. ПРОБЛЕМА УЗКИХ МЕСТ

Задачей об узких местах в зарубежной технической литературе называют задачу о наиболее эффективном распределении ограниченных ресурсов между взаимно связанными звеньями системы. Типичной задачей этого класса является задача об обеспечении возможно более высокой боеготовности противовоздушной обороны в течение заданного промежутка времени [50]. Суть этой задачи заключается в следующем.

Истребители являются эффективным средством ПВО только в том случае, если в войсках имеется достаточное количество подготовленных экипажей. Подготовка экипажей в учебных частях требует времени и самолетов. Следовательно, выпускаемые промышленностью истребители должны направляться частью в войска, частью на полигоны или в учебные подразделения.

Пусть программа выпуска самолетов рассчитана на n месяцев, так что в течение каждого месяца выпускается соответственно

$$a_1, a_2, \dots, a_n$$

истребителей. Если x_j — число самолетов, направленных в j -м месяце в войска, то $(a_j - x_j)$ — число истребителей, посланных в этом месяце для обучения экипажей. Таким образом, z_j — число машин, используемых в j -м месяце программы для обучения экипажей, равно

$$\begin{aligned} z_j &= z_{j-1} + (a_j - x_j), \\ j &= 1, 2, \dots, n. \end{aligned}$$

Предполагается, что число самолетов, используемых для тренировки к моменту начала программы (z_0), задано.

Пусть на каждом самолете, направленном в учебную часть, в течение месяца обучится k экипажей. Следовательно, количество подготовленных экипажей увеличится за месяц на число, превышающее в k раз число самолетов, использованных в предыдущем месяце в качестве тренировочных машин. Обозначая через y_j число подготовленных экипажей, имеющих в наличии в j -м месяце, получим

$$\begin{aligned} y_j &= y_{j-1} + kz_{j-1}, \\ j &= 1, 2, \dots, n. \end{aligned}$$

При этом исходное число подготовленных экипажей к началу программы (y_0) следует считать заданным.

Еще одна группа ограничений на искомые переменные вытекает из тех соображений, что общее количество поступающих в войска истребителей на каждом этапе не должно превышать число подготовленных к этому моменту экипажей, т. е.

$$\begin{aligned} x_1 + x_2 + \dots + x_j &\leq y_j, \\ j &= 1, 2, \dots, n. \end{aligned}$$

Ясно, что все переменные x_j, y_j, z_j неотрицательны:

$$x_j \geq 0, y_j \geq 0, z_j \geq 0 \quad (j = 1, 2, \dots, n).$$

Боеготовность определяется количеством самолетов с подготовленными экипажами и временем их пребывания в войсках, т. е. числом активных самолето-месяцев. Истребители, направленные в войска в первый месяц осуществления программы, будут там находиться в течение всего периода, на который рассчитана программа. Следовательно, первая группа истребителей обеспечивает nx_1 активных самолето-месяцев. Во втором месяце в войска будет направлено x_2 истребителей, которые до конца осуществления программы будут там находиться $n - 1$ месяц и обеспечат $(n - 1)x_2$ активных самолето-месяцев, и т. д.

Таким образом, боеготовность ПВО естественно характеризовать общим числом активных самолето-месяцев

$$nx_1 + (n - 1)x_2 + \dots + 2x_{n-1} + x_n.$$

Как видим, задача об эффективном распределении выпускаемых промышленностью истребителей между войсками и учебными центрами оказывается задачей линейного программирования.

Собирая все условия вместе, запишем задачу в следующем виде.

Требуется обратить в максимум линейную форму

$$L = nx_1 + (n - 1)x_2 + \dots + 2x_{n-1} + x_n$$

при условиях

$$\left. \begin{aligned} x_j + z_j - z_{j-1} &= a_j, \\ y_j - y_{j-1} - kz_{j-1} &= 0, \\ x_1 + x_2 + \dots + x_j - y_j + s_j &= 0, \\ x_j \geq 0; y_j \geq 0; z_j \geq 0; s_j \geq 0; \\ j &= 1, 2, \dots, n. \end{aligned} \right\} \quad (2.34)$$

Переменные s_j — это дополнительные переменные, введенные для того, чтобы задачу можно было записать в канонической форме.

В задаче $4n$ переменных (x_j, y_j, z_j, s_j) и $3n$ условий-ограничений. Векторы условий и вектор ограничений имеют здесь следующий вид

N_0	$A_j^{(x)}$	$A_j^{(y)}$	$A_j^{(z)}$	$A_j^{(s)}$	B
1	0	0	0	0	a_1
2	0	0	0	0	a_2
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$j-1$	0	0	0	0	a_{j-1}
j	1	0	1	0	a_j
$j+1$	0	0	-1	0	a_{j+1}
$j+2$	0	0	0	0	a_{j+2}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
n	0	0	0	0	a_n
$n+1$	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$n+j-1$	0	0	0	0	0
$n+j$	0	1	0	0	0
$n+j+1$	0	-1	$-k$	0	0
$n+j+2$	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2n$	0	0	0	0	0
$2n+1$	0	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$2n+j-1$	0	0	0	0	0

$\mathbb{N}$	$A_j^{(x)}$	$A_j^{(y)}$	$A_j^{(z)}$	$A_j^{(s)}$	B
$2n + j$	1	-1	0	1	0
$2n + j + 1$	1	0	0	0	0
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$3n$	1	0	0	0	0

Заметим, что к аналогичным векторам условий приводят и различные производственные задачи.

В качестве подобной задачи может быть рассмотрена, например, следующая.

Исходное сырье может быть использовано либо целиком для выработки окончательного продукта, либо частично на средства производства и частью для получения окончательного продукта. Увеличение станочного парка форсирует известным образом производство окончательного продукта. Пусть известен график поступления сырья на каждом этапе и общее число этапов, на которое рассчитана производственная программа. Возникает вопрос, как при этих условиях распределять на каждом этапе сырье между производством окончательного продукта и изготовлением промежуточной аппаратуры, чтобы обеспечить максимальный выпуск продукции к заданному сроку. Часто вместо сырья распределению подлежат капиталовложения. И в том и в другом случае мы приходим к задаче линейного программирования с условиями типа (2.34).

К естественному обобщению задач рассматриваемого класса приводят вопросы более сложного многоэтапного планирования, когда сырье, капиталовложения и другие ресурсы распределяются между производством окончательного продукта и рядом промежуточных потребителей, каждый из которых определенным образом форсирует выпуск окончательного продукта.

§ 13. ЗАДАЧА ЦЕЛЕРАСПРЕДЕЛЕНИЙ

В различных военных задачах, связанных с централизованным управлением боем, возникает вопрос о том, как, имея достаточно полную информацию о распределении сил противника, распределить свои силы, чтобы обеспечить максимальный боевой успех. Такие задачи появляются при выборе направления главного удара и вспомогательных операций, при разработке графика налета бомбардировщиков на объекты противника, при постановке боевого задания средствам ПВО и т. д.

В [55] исследована такая задача применительно к целераспределению в ПВО. Задача сформулирована следующим образом.

В границах действия соединения ПВО обнаружено n воздушных целей. Соединение располагает m активными средствами борьбы с воздушным противником. Информация о целях позволяет установить «коэффициенты важности» целей a_j ($j = 1, 2, \dots, n$), дающие основание предпочитать при целераспределении одни цели другим. Если такой информации нет, все a_j полагаются равными единице. Взаимное расположение целей и средств ПВО, уязвимость целей и тактико-технические характеристики средств противовоздушной обороны определяют таблицу чисел p_{ij} ($i = 1, 2, \dots, m; j = 1, 2, \dots, n$) — вероятности поражения j -й цели i -м средством.

Задача состоит в таком распределении целей между всеми средствами ПВО, при котором суммарный ущерб, нанесенный воздушному противнику, был бы наибольшим. Или, точнее, требуется привести выделенные средства ПВО в такое соответствие с целями, чтобы математическое ожидание числа пораженных целей с учетом их важности достигло максимума.

Переведем задачу на формальный язык.

Назовем параметрами управления числа x_{ij} , равные единице, если i -е средство направляется на j -ю цель, и нулю, если i -е средство не должно действовать против j -й цели.

Вероятность того, что j -я цель не будет поражена всеми m средствами (как вероятность совмещения m независимых событий) может быть записана в виде следующего произведения

$$(1 - p_{1j})^{x_{1j}} (1 - p_{2j})^{x_{2j}} \dots (1 - p_{mj})^{x_{mj}} = \prod_{i=1}^m (1 - p_{ij})^{x_{ij}}.$$

Математическое ожидание числа непораженных целей, взвешенное с учетом важности целей, представляет собой сумму

$$M = \sum_{j=1}^n a_j \prod_{i=1}^m (1 - p_{ij})^{x_{ij}},$$

где a_j — коэффициент важности j -й цели.

Задача заключается в выборе такой системы чисел x_{ij} (нулей и единиц), при котором M будет минимальным. При этом естественно соблюдать условия

$$\sum_{j=1}^n x_{ij} = 1 \quad (2.35)$$

для всех i ($i = 1, 2, \dots, m$).

Смысл ограничений (2.35) в том, что каждое средство из наряда, выделенного для борьбы с зафиксированным воздушным противником, обязательно используется.

M представляет собой математическое ожидание числа пропущенных целей с учетом их важности.

Отметим некоторую условность принятой постановки задачи о целераспределении. Здесь предполагается наличие полной информации о воздушной обстановке. Как правило же, работа системы ПВО происходит в условиях, когда цели обнаруживаются последовательно, а не одновременно.

Показатель качества целераспределения M нелинейно зависит от параметров управления x_{ij} . Поэтому поставленная задача не является задачей линейного программирования. Однако в одном важном частном случае, который мы сейчас рассмотрим, решение задачи может быть достигнуто методами линейного программирования. Здесь имеется в виду случай, когда p_{ij} — вероятность поражения j -й цели i -м средством — не зависит от номера средства

$$p_{ij} = p_j.$$

С таким случаем мы сталкиваемся, когда имеются однородные средства ПВО, расположенные компактно так, что зоны досягаемости всех средств мало различаются между собой.

В этом случае показатель качества управления активными средствами ПВО записывается в виде

$$M = \sum_{j=1}^n a_j \prod_{i=1}^m (1 - p_j)^{x_{ij}} = \sum_{j=1}^n a_j (1 - p_j)^{\sum_{i=1}^m x_{ij}}.$$

Введем обозначение

$$x_j = \sum_{i=1}^m x_{ij}.$$

x_j — это количество средств ПВО, направленных против j -й цели. Следовательно, x_j — по условию целое неотрицательное число.

В новых обозначениях

$$M = \sum_{j=1}^n a_j (1 - p_j)^{x_j}. \quad (2.36)$$

Система ограничений (2.35) в принятых предположениях заменяется одним условием

$$\sum_{j=1}^n x_j = \sum_{j=1}^n \left(\sum_{i=1}^m x_{ij} \right) = \sum_{i=1}^m \left(\sum_{j=1}^n x_{ij} \right),$$

или, в силу (2.35),

$$\sum_{j=1}^n x_j = m. \quad (2.37)$$

Смысл этого условия в том, что для борьбы с целями используются все m средств.

Покажем, что определение минимума M в целых точках (а только это нас здесь и интересует) сводится к решению задачи линейного программирования.

Здесь так же, как и в § 9, используется тот факт, что показатель качества решения задачи может быть

представлен в виде суммы выпуклых функций от каждой из переменных

$$M = \sum_{j=1}^n \varphi_j(x_j),$$

где $\varphi_j(x_j) = a_j(1 - p_j)^{x_j}$, как легко убедиться, — выпуклая функция от x_j .

Однако если в § 9 выпуклость $\varphi_j(x_j)$ обеспечила только приближенное решение задачи, то в этом случае можно получить точное решение задачи целераспределения в принятой постановке.

Заменяем функцию $\varphi_j(x_j)$ непрерывной функцией $\tilde{\varphi}_j(x_j)$, совпадающей с $\varphi_j(x_j)$ при целых значениях x_j и линейно изменяющейся между соседними целочисленными значениями x_j . График $\tilde{\varphi}_j(x_j)$ — ломаная линия.

Повторяя рассуждения § 9, заменяя где нужно значения Δ_{ij} целочисленными значениями x_j , придем к следующему выводу.

Две сформулированные ниже задачи имеют одно и то же решение.

Задача 1°. Требуется найти минимум

$$\tilde{M} = \sum_{j=1}^n \tilde{\varphi}_j(x_j)$$

при условиях

$$\sum_{j=1}^n x_j = m, \quad x_j \geq 0 \quad (j = 1, 2, \dots, n).$$

Задача 2°. Требуется вычислить минимум линейной формы

$$L = \sum_{j=1}^n y_j \tag{2.38}$$

при условиях

$$y_j \geq \lambda_{ij}x_j + \mu_{ij}, \tag{2.39}$$

$$\sum_{j=1}^n x_j = m, \tag{2.37}$$

$$x_j \geq 0, \quad (2.40)$$

$$(i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n).$$

Здесь $\lambda_{ij} = \tilde{\varphi}_j(i) - \tilde{\varphi}_j(i-1) = \varphi_j(i) - \varphi_j(i-1)$ — угловой коэффициент наклона i -го звена ломаной;

$$\mu_{ij} = i\tilde{\varphi}_j(i-1) - (i-1)\tilde{\varphi}_j(i) = i\varphi_j(i-1) - (i-1)\varphi_j(i) —$$

отрезок, отсекаемый на оси ординат прямой, содержащей i -е звено ломаной.

Задача 2⁰ является задачей линейного программирования. Докажем теперь, что задача 2⁰ решается в целых числах, т. е. минимум L достигается при целочисленных значениях x_j .

Доказательство основано на следующем результате, доказанном в § 5 главы 1.

Для того чтобы точка $X = (x_1, x_2, \dots, x_n)$ была вершиной многогранника условий, необходимо и достаточно, чтобы ее координаты удовлетворяли по крайней мере n независимым равенствам. Геометрический смысл этого предложения ясен — вершина многогранника в n -мерном пространстве является точкой пересечения n гиперплоскостей. То же предложение в терминах § 8 главы 1 формулируется следующим образом.

План $X = (x_1, x_2, \dots, x_n)$ является опорным в том и только в том случае, если он обращает в равенства не менее чем n независимых условий (считая и условия вида $x_j \geq 0$), где n — число переменных.

В нашей задаче $2n$ переменных — $x_1, x_2, \dots, x_n$ и $y_1, y_2, \dots, y_n$. Следовательно, решение задачи должно удовлетворять по крайней мере $2n$ линейным равенствам.

В силу условий (2.37) и (2.40)

$$0 \leq x_j \leq m.$$

Возможны 2 случая. Если один из x_j равен m , то в силу условий (2.37) и (2.40) все остальные $x_j = 0$, и, следовательно, опорный план состоит из целочисленных значений x_j . Пусть теперь ни один из x_j не равен m , а r значений x_j равны нулю. Нулевое значение x_j и каждое из дробных значений x_j соответствуют только одному из звеньев ломаной $\tilde{\varphi}_j(x_j)$. Поэтому

дробное x_j или $x_j = 0$ могут обратить в равенство не более чем одно из m неравенств вида (2.39)

$$y_j \geq \lambda_{ij}x_j + \mu_{ij};$$

$i = 1, 2, \dots, m$ — номер звена ломаной.

Если же x_j — целое положительное число, то соответствующее y_j может оказаться ординатой точки пересечения двух соседних звеньев ломаной и два из неравенств ограничений вида (2.39) обратятся в равенства. Таким образом, при $x_j > 0$ не более двух неравенств (2.39) (для каждого j) могут обратиться в равенства.

Допустим теперь, что наше утверждение о том, что каждый опорный план рассматриваемой задачи соответствует целочисленным значениям x_j , неверно, и некоторому опорному плану соответствует по крайней мере одно дробное значение x_j . Тогда из условия (2.37) следует, что опорный план содержит не менее двух дробных значений x_j .

Найдем, каким числом ограничено количество условий задачи, которые при нашем допущении могут перейти в равенства.

Принятое предположение о том, что r значений x_j равны нулю, означает, что r условий типа (2.40) перешли в равенства $x_j = 0$ и не более r условий вида (2.39) могут перейти в равенства. Двум дробным значениям x_j может соответствовать не более двух равенств из системы (2.39). Остальные $(n - r - 2)$ значений x_j могут обеспечить переход в равенства не более чем $2(n - r - 2)$ неравенств системы (2.39). Все x_j удовлетворяют, кроме того, равенству (2.37).

Итак, предполагая, что опорный план содержит дробное значение x_j , приходим к выводу, что количество ограничений, которым компоненты опорного плана будут удовлетворять как равенствам, не может превышать

$$r + r + 2 + 2(n - r - 2) + 1 = 2n - 1.$$

Между тем, план может быть опорным только в том случае, если число условий равенств, которым удовлетворяют его компоненты, не меньше числа переменных

в задаче. В рассматриваемой задаче $2n$ переменных. Полученное противоречие свидетельствует о том, что все опорные планы исследуемой задачи состоят только из целочисленных значений x_j . Как известно, в качестве решения задачи линейного программирования может быть принят один из ее оптимальных опорных планов. Поэтому задача 2⁰, а вместе с ней и задача 1⁰ имеют целочисленные решения.

При целых значениях x_j $\varphi_j(x_j)$ совпадает с $\tilde{\varphi}_j(x_j)$. Поэтому решение исходной задачи — вычисление минимума M (2.36) при условиях (2.37) и (2.40) и целочисленных неотрицательных x_j — совпадает с решением задачи линейного программирования с линейной формой (2.38) и условиями (2.37), (2.39) и (2.40).

§ 14. ТЕОРЕТИКО-ИГРОВЫЕ МОДЕЛИ ЗАДАЧ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Военные задачи связаны с изучением конфликтных ситуаций, когда интересы сторон существенно различны, если не противоположны. Для постановки и решения задач подобного рода весьма продуктивными оказываются выводы теории игр.

Теория игр, в частности теория прямоугольных (матричных) игр, тесно связана с линейным программированием. Каждую дискретную игру можно свести к задаче линейного программирования, и, наоборот, каждая задача линейного программирования может быть сведена к эквивалентной прямоугольной игре. Таким образом, методы обоих разделов науки переплетаются, обогащая друг друга.

В § 2 заключения приведены краткие сведения о связи теории игр и линейного программирования. Здесь мы рассмотрим только несколько игровых задач, которые могут быть легко сформулированы в терминах линейного программирования.

Первая из этих задач — это задача о рациональном соотношении между различными типами бронебойных снарядов. В статье [50] рассматривается следующий идеализированный пример.

Пусть известно, что противник использует m различных типов танковой брони $B_1, B_2, \dots, B_m$. Однако неиз-

вестно, какое количество танков оснащено той или иной броней. Имеется n типов бронебойных снарядов, относящихся к классам $C_1, C_2, \dots, C_n$. Известен, кроме того, закон поражения танка, оснащенного тем или иным видом брони, при использовании каждого типа бронебойных снарядов, то есть задана таблица чисел p_{ij} , указывающих вероятность поражения танка с i -й броней снарядом j -го типа.

Тип брони	Тип снаряда			
	C_1	C_2	$\dots$	C_n
B_1	p_{11}	p_{12}	$\dots$	p_{1n}
B_2	p_{21}	p_{22}	$\dots$	p_{2n}
$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$
B_m	p_{m1}	p_{m2}	$\dots$	p_{mn}

Требуется вычислить, в каком соотношении следует производить и запасать бронебойные снаряды различных типов, чтобы обеспечить максимальное значение математического ожидания числа выведенных из строя танков.

Поскольку неизвестно, в каком соотношении противник использует для своих танков броню разных марок, естественно предположить, что он это сделает наилучшим для себя образом. Иными словами, целесообразно считать, что противник, зная, какими типами бронебойных снарядов мы располагаем, будет исходить в производстве брони разных марок из стремления свести к минимуму возможный ущерб, который может быть нанесен его бронетанковым силам.

Обозначим через x_j долю j -го типа снарядов в производстве боеприпасов. Ясно, что

$$\begin{aligned}
 &x_1 + x_2 + \dots + x_n = 1, \\
 &x_j \geq 0 \quad (j = 1, 2, \dots, n).
 \end{aligned}$$

Если снаряды смешивать в соотношении $x_1 : x_2 : \dots : x_n$ и использовать случайным образом, то математическое ожидание числа выведенных из строя танков с броней марки B_1 , отнесенное к одному снаряду, равно

$$M_1 = p_{11}x_1 + p_{12}x_2 + \dots + p_{1n}x_n.$$

При танках с броней B_2

$$M_2 = p_{21}x_1 + p_{22}x_2 + \dots + p_{2n}x_n.$$

И, наконец, при танках с броней марки B_m

$$M_m = p_{m1}x_1 + p_{m2}x_2 + \dots + p_{mn}x_n.$$

Ясно, что математическое ожидание числа уничтоженных танков, отнесенное к одному снаряду, будет равно, по крайней мере, наименьшему из чисел $M_1, M_2, \dots, M_m$.

Обозначим это наименьшее число через x_{n+1}

$$x_{n+1} = \min M_i, \quad i = 1, 2, \dots, m.$$

Величина x_{n+1} представляет собой гарантированное среднее число выведенных из строя танков, отнесенное к одному снаряду (в расчете на разумного противника).

Поэтому числа $x_1, x_2, \dots, x_n$ целесообразно выбирать таким образом, чтобы обеспечить максимальное значение x_{n+1} , учитывая условия

$$x_{n+1} \leq M_i \quad (i = 1, 2, \dots, m),$$

$$x_1 + x_2 + \dots + x_n = 1,$$

$$x_j \geq 0 \quad (j = 1, 2, \dots, n).$$

Таким образом, решение вопроса о выборе рационального соотношения между типами бронебойных снарядов свелось к решению следующей задачи линейного программирования с $(n+1)$ переменными.

Требуется обратить в максимум линейную форму

$$L = x_{n+1}$$

при условиях

[illegible]

К подобным теоретико-игровым моделям сводятся и многие другие военные задачи.

При выборе рационального метода целераспределения мы в § 13 молчаливо предполагали, что замысел воздушного противника известен командованию ПВО. Только при таком допущении имеются основания определять район встречи целей и активных средств ПВО и оценивать характеристики p_{ij} — вероятности поражения j -й цели i -м средством. Такая постановка задачи не всегда может быть признана оправданной. При управлении средствами ПВО естественно предполагать всяческое противодействие противника, снижающее эффективность обороны. Задача воздушного противника заключается в нанесении возможно большего ущерба обороняемому району. Из этих соображений, по-видимому, и следует определять замысел налета.

Для того чтобы формализовать приведенные рассуждения, целесообразно придать каждому обороняемому объекту весовой коэффициент, характеризующий его ценность *). Естественно предположить, что воздушный противник выберет объекты бомбометания (а следовательно, и направления полета) таким образом, чтобы добиться возможно большего средневзвешенного числа уничтоженных объектов. Задача противовоздушной обороны прямо противоположна. Активные средства ПВО должны быть так распределены по целям, чтобы свести к минимуму возможный ущерб обороняемому району.

*) Выбор характеристик ценности объекта представляет собой самостоятельную весьма сложную задачу. Мы ее, конечно, здесь не можем касаться.

Выбор варианта налета и параметров управления системой ПВО при той или иной информации у обеих сторон — предметы стратегии двух противников — средств воздушного нападения и противовоздушной обороны.

В простейшем случае постановка и решение такой задачи мало отличаются от рассмотренной выше задачи о выборе рационального соотношения между различными видами бронейных боеприпасов.

Такова же схема исследования задач о рациональном выборе калибров, об определении диапазона частот для работы радиолокаторов в условиях активных помех, о соотношении между средствами нападения и средствами обороны и т. д.

Во всех этих задачах наибольшую трудность представляет собой выбор критериев качества и ограничений, которым подчиняются стратегии противников.

ГЛАВА 3

МЕТОД ПОСЛЕДОВАТЕЛЬНОГО УЛУЧШЕНИЯ ПЛАНА

В этой главе излагается один из общих приемов линейного программирования. Основы метода были сформулированы Данцигом в 1949 г. [28], [29]. В иностранной литературе по линейному программированию метод Данцига известен под названием симплексного метода. Такое название возникло из геометрического истолкования первых частных задач, к которым он был применен, и не соответствует существу метода.

Идея метода содержит три существенных момента. Во-первых, указывается способ вычисления опорного плана. Во-вторых, устанавливается критерий, который позволяет проверить, является ли выбранный опорный план оптимальным (или в принятой терминологии — является ли выбранный план решением). В-третьих, приводится способ, позволяющий по выбранному неоптимальному опорному плану получить другой опорный план, более близкий к оптимальному. Доказывается, что таким образом можно через конечное число шагов получить оптимальный план — решение задачи линейного программирования. Таким образом, метод заключается в последовательном улучшении плана, что и целесообразно отразить в его названии.

Следует заметить, что алгоритмы метода позволяют также в процессе вычислений установить, является ли задача линейного программирования разрешимой. Это значит, что в ходе расчетов можно определить, не оказываются ли условия задачи противоречивыми и обеспечивают ли они ограниченность линейной формы задачи.

Описание метода последовательного улучшения плана проводится здесь по следующей схеме. Вначале излагаются основы метода применительно к так называемому невырожденному случаю. При этом предполагается известным некоторый опорный план исследуемой задачи. Затем подробно описывается способ построения исходного опорного плана.

В последующих параграфах приводится изложение двух различных алгоритмов метода. После этого мы освобождаемся от ограничивающего допущения о невырожденности рассматриваемой задачи.

§ 1. ОСНОВЫ МЕТОДА

1.1. Запишем задачу линейного программирования в канонической форме.

Требуется вычислить максимум линейной формы

$$L(X) = c_1x_1 + c_2x_2 + \dots + c_nx_n \quad (3.1)$$

при условиях

$$A_1x_1 + A_2x_2 + \dots + A_nx_n = B, \quad (3.2)$$

$$x_j \geq 0 \quad (j = 1, 2, \dots, n). \quad (3.3)$$

Здесь A_j и B соответственно векторы условий и вектор ограничений

$$A_j = (a_{1j}, a_{2j}, \dots, a_{mj});$$

$$B = (b_1, b_2, \dots, b_m).$$

В этом и в последующих параграфах вплоть до § 7 все встречающиеся в ходе изложения задачи линейного программирования предполагаются невырожденными. Приведем определение невырожденной задачи.

Задача линейного программирования, записанная в канонической форме (3.1)—(3.3) называется невырожденной, если каждый ее опорный план содержит ровно m положительных компонент.

Напомним, что система векторов условий, отвечающих положительным составляющим опорного плана, была названа в гл. 1 базисом этого плана.

Перейдем теперь к изложению основ метода последовательного улучшения плана.

Допустим, что нам известен некоторый опорный план $X = (x_1, x_2, \dots, x_m, \dots, x_n)$ задачи (3.1) — (3.3). Для определенности будем считать его базис составленным из первых m векторов условий $A_1, A_2, \dots, A_m$. К этому всегда можно прийти, изменив соответствующим образом нумерацию переменных задачи. Таким образом, компоненты опорного плана X с номерами, превышающими m , равны нулю:

$$x_{m+1} = \dots = x_n = 0.$$

Зафиксируем теперь произвольный вектор условий A_j из числа векторов, не входящих в базис плана X . Проследим, как изменится линейная форма (3.1) в результате перехода к некоторому новому плану $X(\theta)$, который образуется из плана X в соответствии со следующим правилом.

j -я компонента плана $X(\theta)$ полагается равной некоторому положительному числу θ

$$x_j(\theta) = \theta.$$

Составляющие при остальных векторах условий, не входящих в базис плана X , остаются равными нулю

$$x_{m+1}(\theta) = \dots = x_{j-1}(\theta) = x_{j+1}(\theta) = \dots = x_n(\theta) = 0.$$

Что касается первых m компонент $X(\theta)$, то они выбираются так, чтобы система чисел $x_1(\theta), x_2(\theta), \dots, x_n(\theta)$ определяла план задачи, т. е. удовлетворяла условиям (3.2) — (3.3).

Последние $n - m$ компонент плана X равны нулю. Поэтому

$$A_1 x_1 + A_2 x_2 + \dots + A_m x_m = B. \quad (3.4)$$

Исходный план X придает линейной форме (3.1) значение

$$L(X) = c_1 x_1 + \dots + c_m x_m. \quad (3.5)$$

В силу предположений о структуре плана $X(\theta)$ его отличные от нуля компоненты должны удовлетворять векторному равенству

$$\sum_{i=1}^m x_i(\theta) A_i + \theta A_j = B$$

или

$$\sum_{i=1}^m x_i(\theta) A_i = B - \theta A_j. \quad (3.6)$$

Векторное уравнение (3.6) можно рассматривать, как систему m линейных уравнений относительно неизвестных $x_1(\theta), x_2(\theta), \dots, x_m(\theta)$. Определитель системы (3.6) отличен от нуля, так как векторы $A_1, A_2, \dots, A_m$ образуют базис опорного плана. Поэтому неизвестные составляющие плана $X(\theta)$ определяются единственным образом. Свяжем компоненты $x_i(\theta)$ с составляющими x_i исходного опорного плана. Воспользуемся для этого следующим приемом. Выпишем разложение вектора A_j по векторам $A_1, A_2, \dots, A_m$

$$A_j = \sum_{i=1}^m z_{ij} A_i. \quad (3.7)$$

Умножим (3.7) на θ и вычтем результат из (3.4). Имеем

$$\sum_{i=1}^m (x_i - \theta z_{ij}) A_i = B - \theta A_j.$$

Сравнивая полученные соотношения с системой (3.6) и учитывая единственность ее решения, приходим к равенствам

$$x_i(\theta) = x_i - \theta z_{ij}, \quad i = 1, 2, \dots, m. \quad (3.8)$$

Будем предполагать θ настолько малым, чтобы все величины из (3.8) были неотрицательными. (Напомним, что x_i ($i = 1, 2, \dots, m$) — положительные компоненты плана X). Тогда $X(\theta)$ удовлетворяет условиям (3.2) и (3.3) и является планом рассматриваемой задачи линейного программирования. Зная все составляющие нового плана $X(\theta)$, нетрудно вычислить соответствующее ему значение линейной формы (3.1)

$$\begin{aligned} L[X(\theta)] &= \sum_{i=1}^m c_i x_i(\theta) + c_j \theta = \\ &= \sum_{i=1}^m c_i (x_i - \theta z_{ij}) + c_j \theta = \sum_{i=1}^m c_i x_i + \theta \left(c_j - \sum_{i=1}^m c_i z_{ij} \right). \end{aligned}$$

Введем обозначения

$$z_j = \sum_{i=1}^m c_i z_{ij}, \quad (3.9)$$

$$\Delta_j = z_j - c_j, \quad (3.10)$$

неоднократно используемые в дальнейшем изложении.

В новых обозначениях выражение для $L[X(\theta)]$ принимает следующий вид:

$$L[X(\theta)] = L(X) - \theta \Delta_j. \quad (3.11)$$

Соотношение (3.11) показывает, что влияние рассмотренного преобразования исходного плана X на величину линейной формы (3.1) определяется знаком Δ_j . Если $\Delta_j > 0$, линейная форма уменьшается; если $\Delta_j < 0$, она увеличивается; если $\Delta_j = 0$, линейная форма сохраняет свое прежнее значение.

Рассмотренное преобразование опорного плана (назовем его элементарным преобразованием) носит достаточно частный характер. Тем не менее с помощью элементарных преобразований всегда может быть осуществлено построение оптимального плана исследуемой задачи. Решение задачи линейного программирования методом последовательного улучшения плана как раз и складывается из определенной последовательности этих преобразований.

1.2. Допустим, что в соответствии с формулами (3.7), (3.9) и (3.10) мы вычислили величины z_{ij} и Δ_j для $i = 1, 2, \dots, m$; $j = m+1, m+2, \dots, n$. При этом может иметь место один из следующих трех случаев:

1°. $\Delta_j \geq 0$ для $j = m+1, m+2, \dots, n$. (Ясно, что при $j = 1, 2, \dots, m$ $z_j = c_j$. Поэтому в случае 1° $\Delta_j \geq 0$ для всех j от 1 до n).

2°. $\Delta_j < 0$ для некоторого j и все соответствующие этому индексу величины $z_{ij} \leq 0$ ($i = 1, 2, \dots, m$).

3°. $\Delta_j < 0$ для некоторых индексов j , и для каждого такого j по крайней мере одно из чисел z_{ij} положительно.

Разберем подробно каждую из указанных возможностей.

1⁰. Как мы видели в предыдущем пункте, условия, составляющие содержание первого случая, свидетельствуют о том, что ни одно из элементарных преобразований не приводит к увеличению линейной формы задачи. Мы сейчас покажем, что отсюда следует оптимальность опорного плана X . Соответствующее утверждение, называемое обычно критерием оптимальности, является одним из основных принципов метода последовательного улучшения плана.

Критерий оптимальности. Если $\Delta_j \geq 0$ для $j = m+1, m+2, \dots, n$, план $X = (x_1, x_2, \dots, x_m, 0, \dots, 0)$ является решением задачи (3.1) — (3.3).

Докажем справедливость сформулированного утверждения.

Рассмотрим любой план $\tilde{X} = (\tilde{x}_1, \tilde{x}_2, \dots, \tilde{x}_n) \neq X$. Как и компоненты всякого плана, составляющие $\tilde{X}$, должны удовлетворять условиям задачи, т. е.

$$B = \sum_{j=1}^n \tilde{x}_j A_j,$$

$$\tilde{x}_j \geq 0, \quad j = 1, 2, \dots, n.$$

Разложим векторы A_j согласно формуле (3.7) по векторам базиса, определяемого опорным планом X . Получим

$$B = \sum_{j=1}^n \tilde{x}_j \left(\sum_{i=1}^m z_{ij} A_i \right)$$

или, меняя порядок суммирования,

$$B = \sum_{i=1}^m \left(\sum_{j=1}^n \tilde{x}_j z_{ij} \right) A_i. \quad (3.12)$$

Векторы $A_1, A_2, \dots, A_m$ составляют линейно независимую систему. Следовательно, вектор ограничений B может быть единственным образом представлен как линейная комбинация этих векторов. Сравнивая (3.4) и (3.12), заключаем, что

$$\sum_{j=1}^n \tilde{x}_j z_{ij} = x_i. \quad (3.13)$$

Плану $\tilde{X}$ соответствует значение линейной формы

$$L(\tilde{X}) = \sum_{j=1}^n c_j \tilde{x}_j.$$

По условию $z_j - c_j = \Delta_j \geq 0$ для всех j . Следовательно,

$$L(\tilde{X}) = \sum_{j=1}^n c_j \tilde{x}_j \leq \sum_{j=1}^n z_j \tilde{x}_j.$$

Подставляя в эту формулу значение z_j из (3.9) и меняя порядок суммирования, приходим к неравенству

$$L(\tilde{X}) \leq \sum_{j=1}^n \left(\sum_{i=1}^m c_i z_{ij} \right) \tilde{x}_j = \sum_{i=1}^m \left(\sum_{j=1}^n \tilde{x}_j z_{ij} \right) c_i.$$

Учитывая далее (3.13), получим требуемый результат

$$L(\tilde{X}) \leq \sum_{i=1}^m x_i c_i = L(X).$$

Таким образом, при $\Delta_j \geq 0$ для всех j (случай 1⁰) опорный план X определяет максимальное значение линейной формы и является решением задачи линейного программирования.

2⁰. В этом случае имеется индекс j , для которого $\Delta_j < 0$ и все соответствующие компоненты $z_{ij} \leq 0$ ($i = 1, 2, \dots, m$).

Применим к плану X элементарное преобразование, связанное с вектором A_j . Очевидно, что при любом положительном θ

$$x_i(\theta) = x_i - \theta z_{ij} \geq 0 \quad (i = 1, 2, \dots, m),$$

т. е. $X(\theta)$ является планом задачи (3.1) — (3.3). Формула (3.11) показывает, что в этом случае линейная форма (3.1) неограничена сверху на множестве планов задачи.

Таким образом, в случае 2⁰ задача линейного программирования неразрешима.

Остается рассмотреть третью возможность.

3⁰. В этом случае $\Delta_j < 0$ для некоторых j и для каждого такого j по крайней мере одно из чисел z_{ij} положительно. Покажем, что здесь можно при помощи эле-

ментарного преобразования, связанного с вектором A_j , перейти от опорного плана X к новому опорному плану X' и увеличить при этом значение линейной формы задачи.

Выберем в качестве θ наименьшее значение отношения $\frac{x_i}{z_{ij}}$ по всем i из множества $1, 2, \dots, m$, для которых $z_{ij} > 0$ (по условию такие z_{ij} имеются).

Итак, пусть

$$\theta_0 = \min_i \frac{x_i}{z_{ij}}.$$

Заметим сразу же, что в силу положительности первых m компонент плана X значение θ_0 положительно. Здесь мы существенно используем невырожденность рассматриваемой задачи. Из невырожденности следует также, что минимум отношения $\frac{x_i}{z_{ij}}$ достигается только при одном значении индекса i . (В противном случае, как это следует из формул (3.6) и (3.8), вектор ограничений B выражался бы менее чем через m векторов условий A_j).

Положим $X' = X(\theta_0)$. По построению вектор $X(\theta)$, полученный путем элементарного преобразования вектора X , удовлетворяет условиям (3.2) при любом θ . Кроме того, из соотношений (3.8) следует, что в рассматриваемом случае при $\theta = \theta_0$ все компоненты x_i' вектора X неотрицательны. Следовательно, $X' = X(\theta_0)$ является планом задачи (3.1) — (3.3). Более того, мы сейчас установим, что X' является опорным планом задачи.

Пусть минимум отношения $\frac{x_i}{z_{ij}}$, определяющий число θ_0 , достигается при $i = r$. Очевидно,

$$x_r' = x_r(\theta_0) = 0.$$

В силу (3.6) при выбранном $\theta = \theta_0$ вектор ограничений B оказывается линейной комбинацией векторов $A_1, A_2, \dots, A_{r-1}, A_{r+1}, \dots, A_m, A_j$ с положительными коэффициентами. Чтобы установить опорность плана X , достаточно убедиться в том, что указанная система векторов линейно независима.

Рассмотрим произвольную линейную комбинацию векторов $A_1, A_2, \dots, A_{r-1}, A_{r+1}, \dots, A_m, A_j$, равную нулю

$$\alpha_1 A_1 + \alpha_2 A_2 + \dots + \alpha_{r-1} A_{r-1} + \alpha_{r+1} A_{r+1} + \dots + \alpha_m A_m + \alpha_j A_j = 0.$$

Умножим обе части равенства (3.7) на α_j и сложим почленно с имеющимся соотношением. Получим

$$(\alpha_1 + \alpha_j z_{1j}) A_1 + \dots + (\alpha_{r-1} + \alpha_j z_{r-1,j}) A_{r-1} + \alpha_j z_{rj} A_r + \dots + (\alpha_{r+1} + \alpha_j z_{r+1,j}) A_{r+1} + \dots + (\alpha_m + \alpha_j z_{mj}) A_m = 0.$$

В силу линейной независимости системы векторов $A_1, A_2, \dots, A_m$ (они составляют базис опорного плана X) имеем

$$\alpha_i + \alpha_j z_{ij} = 0 \quad \text{при } i = 1, 2, \dots, r-1, r+1, \dots, m.$$

$$\alpha_j z_{rj} = 0.$$

Индекс r выбран среди индексов i , для которых z_{ij} положительны. Поэтому $z_{rj} > 0$. Но тогда $\alpha_j = 0$, и, следовательно,

$$\alpha_1 = \alpha_2 = \dots = \alpha_{r-1} = \alpha_{r+1} = \dots = \alpha_m = 0.$$

Таким образом, произвольная линейная комбинация векторов $A_1, \dots, A_{r-1}, A_{r+1}, \dots, A_m, A_j$ может равняться нулю лишь при нулевых коэффициентах. По определению это означает линейную независимость системы векторов.

Следовательно, план X' действительно является опорным планом задачи и его базис состоит из векторов

$$A_1, A_2, \dots, A_{r-1}, A_{r+1}, \dots, A_m, A_j.$$

В соответствии с соотношением (3.11)

$$L(X') = L(X) - \theta_0 \Delta_j.$$

По условию $\Delta_j < 0$. Кроме того, как мы видели, величина θ_0 положительна. Поэтому

$$L(X') > L(X).$$

Таким образом, в случае 3^0 можно перейти от исходного опорного плана X к новому опорному плану X' , более близкому к решению задачи. При этом базис нового плана образуется из предшествующего базиса заменой одного из его векторов (A) вектором A_j , для которого $\Delta_j < 0$. Переход к новому плану приводит к увеличению линейной формы задачи на величину $|\theta_0 \Delta_j|$.

1.3. Повторим кратко порядок операций, связанных с отдельным шагом (с отдельной итерацией) метода последовательного улучшения плана.

Пусть имеется некоторый опорный план задачи X . Решив системы уравнений (3.7) для $j = m+1, \dots, n$, определим числа z_{ij} . С помощью них составляем по формулам (3.9), (3.10) величины Δ_j ($j = m+1, m+2, \dots, n$).

Значения Δ_j и z_{ij} позволяют установить, какой из трех рассмотренных случаев имеет место.

В случае 1^0 решение закончено: X является оптимальным планом задачи линейного программирования. В случае 2^0 процесс вычислений должен быть прекращен — задача неразрешима. Если же имеет место случай 3^0 , то, как мы видели, можно указать способ определения нового опорного плана, увеличивающего значение линейной формы.

Читатель, наверное, обратил внимание на то, что наиболее трудоемкая часть каждой итерации состоит в вычислении величин z_{ij} . До сих пор мы связывали определение этих чисел с необходимостью решения $n - m$ систем линейных уравнений, каждая из которых имеет порядок m . При описании алгоритма метода последовательного улучшения плана будет показано, что величины z_{ij} отвечающие двум соседним шагам метода, связаны простыми рекуррентными соотношениями. Таким образом, удастся избежать трудоемкого процесса решения многих систем линейных уравнений.

Переход от одного опорного плана к следующему осуществляется до тех пор, пока либо не будет получено

решение задачи, либо не будет установлена ее неразрешимость.

Каждый переход от одного опорного плана к следующему — шаг (итерация) метода последовательного улучшения плана. Число шагов, в результате которых мы приходим к решению невырожденной задачи или доказываем ее неразрешимость, конечно. Действительно, каждому опорному плану невырожденной задачи соответствует своя система из m линейно независимых векторов условий, составляющих его базис. Общее число векторов условий задачи равно n . Поэтому существует не более C_n^m различных базисов. Таким образом, и число различных опорных планов задачи конечно. Для каждого опорного плана однозначно определяется величина линейной формы. В невырожденном случае каждый следующий шаг увеличивает значение линейной формы. Поэтому не может быть возвращения к ранее пройденному базису. Это значит, что через конечное число шагов — итераций метода последовательного улучшения плана — будет получено решение или установлена неограниченность линейной формы на множестве планов задачи.

1.4. Все элементы метода последовательного улучшения плана находят простое геометрическое истолкование. В § 7 гл. 1 были даны две геометрические интерпретации задачи. Читателю полезно будет вернуться к ним и связать приведенное в настоящем параграфе алгебраическое описание метода с его геометрией. Здесь мы лишь кратко наметим эту связь применительно к первой геометрической интерпретации задачи линейного программирования. Условия задач (3.2), (3.3) отсекают в пространстве переменных $x_1, \dots, x_n$ выпуклый многогранник (или неограниченное выпуклое многогранное тело), размерность которого не превышает $n - m$. Исходный опорный план X соответствует вершине многогранника. Геометрическое место точек, в которых линейная форма (3.1) сохраняет постоянное значение $L = \text{const}$, представляет собой гиперплоскость. Коэффициенты c_i определяют направление, в котором следует смещать гиперплоскость, чтобы увеличить значение линейной формы.

Пусть вершина X соответствует исходному опорному плану. Чтобы проверить, достигает ли линейная форма

своего максимума в точке X , следует среди ребер многогранника, исходящих из вершины X , выделить те, вдоль которых увеличивается значение формы. Если таких ребер нет (т. е., если $\Delta_j \geq 0$ для всех j), то исходный план X является оптимальным. В этом случае многогранник условий расположен по одну сторону от гиперплоскости $L = \text{const}$ («под» ней).

Пусть теперь некоторые из Δ_j оказываются отрицательными, т. е. имеются ребра, исходящие из вершины, вдоль которых значение линейной формы увеличивается. При этом возможны два случая. Некоторые из таких ребер могут оказаться лучами. Это значит, что линейная форма неограничена в области своего определения (случай 2⁰).

Случай 3⁰ геометрически соответствует движению вдоль любого из выделенных ребер (все они являются конечными отрезками) в соседнюю вершину X' . При этом $L(X') > L(X)$. Перемещая таким образом гиперплоскость $L = \text{const}$ параллельно самой себе от одной вершины многогранника условий к соседней, мы увеличиваем от шага к шагу значение линейной формы. Конечное число вершин многогранника условий гарантирует достижение максимума линейной формы за конечное число шагов. Из геометрических соображений ясно также, что неразрешимость задачи (неограниченность линейной формы в области своего определения), если она имеет место, обнаруживается опять-таки через конечное число шагов.

§ 2. ВЫБОР НАЧАЛЬНОГО ОПОРНОГО ПЛАНА

2.1. В рассуждениях предыдущего параграфа мы отпирывались от некоторого начального опорного плана, не оговаривая, каким образом можно такой опорный план построить. Существует несколько различных методов построения исходного опорного плана. Во всех этих методах вычисление опорного плана представляет собой, вообще говоря, работу, не менее громоздкую, чем решение задачи линейного программирования при наличии начального опорного плана.

Рассмотрим один из таких приемов.

Запишем задачу линейного программирования в канонической форме.

Требуется обратить в максимум линейную форму

$$L = c_1 x_1 + c_2 x_2 + \dots + c_n x_n \quad (3.1)$$

при условиях

$$\left. \begin{array}{l} a_{11}x_1 + a_{12}x_2 + \dots + a_{1n}x_n = b_1, \\ .\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad.\quad. \\ a_{m1}x_1 + a_{m2}x_2 + \dots + a_{mn}x_n = b_m, \end{array} \right\} \quad (3.2)$$

$$x_j \geq 0 \quad (j=1, 2, \dots, n). \quad (3.3)$$

Здесь можно считать все $b_i \geq 0$. В противном случае следовало бы в соответствующей строке поменять в обеих частях равенства все знаки на обратные.

Рассмотрим наряду со сформулированной задачей следующую вспомогательную задачу. Требуется обратиться в максимум линейную форму

$$\tilde{L} = -(x_{n+1} + x_{n+2} + \dots + x_{n+m}) \quad (3.14)$$

при условиях

[illegible]

$$x_j \geq 0, \quad j = 1, 2, \dots, n + m. \quad (3.16)$$

Решим вспомогательную задачу методом последовательного улучшения плана. Начальный план вспомогательной задачи может быть непосредственно указан. Он составляется из компонент

$$\begin{aligned} x_j &= 0 && \text{при } j \leq n, \\ x_{n+i} &= b_i && \text{при } i = 1, 2, \dots, m. \end{aligned}$$

Нетрудно видеть, что этот план является опорным. Действительно, определитель, составленный из векторов условий при дополнительных переменных, равен единице. Следовательно, соответствующие векторы условий линейно независимы. Линейная форма L ограничена сверху на множестве своих планов ($L \leq 0$). Поэтому процесс последовательного улучшения плана через конечное

число шагов приведет к оптимальному опорному плану вспомогательной задачи *). Возможны 2 случая.

1°. Оптимальное значение $\tilde{L}$ равно нулю.

2°. Оптимальное значение $\tilde{L}$ отрицательно.

В случае 1° оптимальный план вспомогательной задачи оказывается опорным планом исходной задачи. Действительно, при $L = 0$ все $x_{n+i} = 0$ ($i = 1, 2, \dots, m$). Следовательно, решение вспомогательной задачи удовлетворяет условиям исходной задачи. Оптимальный план вспомогательной задачи совпадает, таким образом, с некоторым планом исходной задачи. По построению этот план является опорным.

В случае 2° исходная задача не имеет ни одного плана, т. е. система условий исходной задачи несовместна. В этом нетрудно убедиться, если допустить противное, т. е. принять, что исходная задача имеет хотя бы один план. При таком допущении вспомогательная задача будет иметь план, у которого первые n составляющих совпадают с соответствующими компонентами исходного плана, а последующие m компонент равны нулю. Значение линейной формы для этого плана равно нулю, что противоречит условию, согласно которому максимальное значение $\tilde{L}$ отрицательно.

Итак, начальный опорный план любой задачи линейного программирования может быть вычислен в результате решения вспомогательной задачи с очевидным начальным опорным планом.

2.2. Далеко не во всех случаях имеет смысл разделять решение задачи линейного программирования на два этапа — вычисление начального опорного плана и определение оптимального плана. Рассмотрим прием, позволяющий объединить оба этапа. Сущность приема заключается в том, что вместо исходной задачи линейного программирования решается расширенная задача, имеющая другие опорные планы (один из них всегда легко указать), но те же решения, т. е. те же оптимальные планы, что и исходная задача. Множество опорных

*) До сих пор это утверждение было доказано лишь для невырожденной задачи, однако в § 7 оно будет перенесено и на вырожденный случай.

планов расширенной задачи содержит в качестве подмножества все опорные планы исходной задачи.

Рассмотрим наряду с исходной задачей линейного программирования следующую расширенную задачу.

Требуется обратить в максимум линейную форму

$$\tilde{L} = c_1 x_1 + \dots + c_n x_n - M(x_{n+1} + \dots + x_{n+m}),$$

где $M > 0$ достаточно большое число, а $X = (x_1, \dots, x_{n+m})$ удовлетворяет условиям (3.15), (3.16). Назовем расширенную задачу M -задачей.

Докажем следующие два утверждения, определяющие идею приема.

I. Если в оптимальном плане $\tilde{X}$ M -задачи $x_{n+i} = 0$ ($i = 1, \dots, m$), т. е. если $\tilde{X} = (x_1, \dots, x_n, 0, \dots, 0)$, то план $X = (x_1, x_2, \dots, x_n)$ является решением исходной задачи.

II. Всегда можно указать такое число $M_0 > 0$, что для любого $M > M_0$ из существования хотя бы одного плана исходной задачи вытекает соотношение

$$x_{n+i} = 0 \quad (i = 1, 2, \dots, m)$$

для оптимального плана M -задачи.

Докажем I утверждение. Если $\tilde{X}$ удовлетворяет условиям расширенной задачи, то X удовлетворяет условиям исходной задачи. Следовательно, X является планом исходной задачи.

Установим его оптимальность. Для этого допустим существование такого плана $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ задачи (3.1)–(3.3), что

$$L(X^*) > L(X).$$

$\tilde{X}^* = (x_1^*, \dots, x_n^*, 0, \dots, 0)$ — план M -задачи. Имеем

$$\tilde{L}(\tilde{X}^*) = L(X^*) > L(X) = \tilde{L}(\tilde{X}).$$

Полученное неравенство противоречит тому, что $\tilde{X}$ является решением M -задачи.

Итак, если в оптимальном плане M -задачи $x_{n+i} = 0$ для $i = 1, 2, \dots, m$, то первые n компонент этого плана определяют решение исходной задачи.

Начальный опорный план M -задачи $\tilde{X}_0$ может быть непосредственно указан

$$\tilde{X}_0 = (\underbrace{0, \dots, 0}_n, b_1, \dots, b_m).$$

Заметим, что базис, составленный из единичных векторов при дополнительных переменных, обычно называют искусственным базисом.

Перейдем к доказательству II утверждения. Допустим, что утверждение неверно, т. е. существует план $X = (x_1, \dots, x_n)$ исходной задачи и в то же время нельзя указать такое число M_0 , что при всех $M > M_0$ для решения M -задачи $x_{n+i} = 0$ для $i = 1, 2, \dots, m$.

Введем числа m_1 и m_2 следующим образом: m_1 равно минимуму суммы $x_{n+1} + \dots + x_{n+m}$ по всем опорным планам M -задачи, которые не обращают эту сумму в нуль (указанный минимум существует, поскольку число опорных планов M -задачи конечно и не зависит от M); m_2 равно максимальному значению линейной формы

$$c_1 x_1 + \dots + c_n x_n$$

на множестве опорных планов M -задачи.

Ясно, что

$$m_2 \geq L(X).$$

Положим

$$M_0 = 2 \frac{m_2 - L(X)}{m_1}$$

и выберем $M > M_0$. Согласно допущению по крайней мере одна из компонент x_{n+i}^* ($i = 1, 2, \dots, m$) оптимального плана $X^* = (x_1^*, \dots, x_n^*, x_{n+1}^*, \dots, x_{n+m}^*)$ M -задачи положительна. Поэтому $m_1 > 0$.

В соответствии с определением m_1 и m_2

$$\sum_{i=1}^m x_{n+i}^* \geq m_1; \quad \sum_{j=1}^n c_j x_j^* \leq m_2.$$

Следовательно,

$$\tilde{L}(X^*) = \sum_{j=1}^n c_j x_j^* - M \sum_{i=1}^m x_{n+i}^* \leq m_2 - M m_1.$$

С другой стороны, дополнив нулями план X исходной задачи, получим план $\tilde{X} = (x_1, \dots, x_n, 0, \dots, 0)$ M -задачи.

При этом

$$\tilde{L}(\tilde{X}) = L(X)$$

и

$$\tilde{L}(\tilde{X}) \leq L(X^*),$$

поскольку X^* решение M -задачи.

Используя последние три соотношения, получаем

$$L(X) \leq m_2 - Mm_1,$$

или (поскольку $m_1 > 0$)

$$M \leq \frac{m_2 - L(X)}{m_1} = \frac{M_0}{2}.$$

Полученное неравенство противоречит условию $M > M_0$. Утверждение II доказано.

Установленные здесь положения обеспечивают возможность решения задачи линейного программирования с неизвестным заранее начальным опорным планом. Исходная задача сводится к M -задаче, для которой выбор начального опорного плана не представляет труда. Такой метод решения задачи линейного программирования называется M -методом. Он весьма эффективен при $m \ll n$.

Для решения задачи нет необходимости в вычислении величины M_0 . При определении значений Δ_j в процессе решения M -задачи следует полагать M больше любого сравниваемого с ним числа. Если в процессе решения M -задачи будет получен оптимальный план, для которого $x_{n+i} = 0$ для $i = 1, 2, \dots, m$, то первые m его компонент определяют решение исходной задачи. Если по крайней мере одна из составляющих x_{n+i} положительна при любом сколь угодно большом M , исходная задача не имеет ни единого плана, т. е. условия задачи не совместимы между собой. Итак, M -метод позволяет вычислить оптимальный план любой разрешимой задачи линейного программирования без предварительного определения начального опорного плана.

Пример использования M -метода будет приведен в § 6.

2.3. Идея введения дополнительных переменных (искусственного базиса) полезна всегда, когда ограничения искомых переменных заданы неравенствами

$$A_1x_1 + A_2x_2 + \dots + A_nx_n \leq B.$$

(Составляющие вектора ограничений B , как и прежде, предполагаются неотрицательными.) В этом случае задача сводится к канонической форме, если ввести m дополнительных неотрицательных переменных $x_{n+1}, \dots, x_{n+m}$ и соответствующие векторы условий $A_{n+1}, \dots, A_{n+m}$. Все компоненты вектора A_{n+k} , за исключением k -й равны нулю; k -я составляющая A_{n+k} равна 1. В линейной форме коэффициенты при дополнительных переменных равны нулю, и, следовательно, величины x_{n+k} не оказывают влияния на значение формы.

Условия задачи, записанные в канонической форме, имеют вид

$$A_1x_1 + \dots + A_nx_n + A_{n+1}x_{n+1} + \dots + A_{n+m}x_{n+m} = B.$$

Матрица

$$(A_{n+1}, \dots, A_{n+m}) = \begin{vmatrix} 1 & 0 \dots 0 \\ 0 & 1 \dots 0 \\ \vdots & \vdots \ddots \vdots \\ 0 & 0 \dots 1 \end{vmatrix}$$

— единичная матрица. Следовательно, дополнительные векторы образуют базис, которому соответствует начальный опорный план задачи

$$X = (0, \underbrace{0, \dots, 0}_n, b_1, b_2, \dots, b_m).$$

Разложение векторов условий A_j по векторам начального единичного базиса не представляет труда. При единичном начальном базисе отпадает необходимость в решении системы уравнений для вычисления z_{ij} .

Отсутствие необходимости в решении систем линейных уравнений существенно упрощает вычисления в методе последовательного улучшения плана. Однако при этом следует иметь в виду, что, выбирая дополнительные векторы в качестве векторов начального базиса, мы исходим обычно из худшего первого приближения, чем в тех случаях, когда в базис с самого начала вводятся

векторы, соответствующие основным переменным. Количество приближений, требуемых для вычисления оптимального плана, при этом, как правило, увеличивается. Тем не менее при использовании универсальных вычислительных машин введение искусственного базиса (соответствующего дополнительным переменным) оправдывает себя. Увеличение количества вычислений, которое может при этом произойти, связано только с дополнительными однообразными весьма простыми операциями.

§ 3. СВЯЗЬ МЕЖДУ ПАРАМЕТРАМИ ПОСЛЕДОВАТЕЛЬНЫХ ПРИБЛИЖЕНИЙ

3.1. Мы видим, что решение задачи линейного программирования методом последовательного улучшения плана, состоит из ряда однотипных шагов. На каждом шаге необходимо вычислить следующие параметры:

а) компоненты z_{ij} разложения всех векторов условий по векторам базиса;

б) составляющие x_i опорного плана, т. е. компоненты разложения вектора ограничений по векторам базиса;

в) величины $\Delta_j = z_j - c_j$, необходимые для оценки плана и перехода к новому базису*);

г) значение линейной формы $L = L(X)$.

Вычисление параметров z_{ij} ($i = 1, 2, \dots, m$) на первом шаге связано, вообще говоря, с решением систем линейных уравнений (3.7). В векторной форме эти системы удобно записывать в виде

$$A_j = \sum_{i \in B_1} z_{ij} A_i. \quad (3.17)$$

Обозначение $i \in B_1$ под знаком суммы означает, что суммирование ведется по индексам i , отвечающим номерам векторов условий, входящих в первый базис (B_1). (Знак $\in$ означает принадлежность. Запись $i \in B_1$ читается: i принадлежит B_1 .) Мы уже видели, что если B_1 состоит из единичных векторов, то необходимость в решении системы уравнений (или, что то же самое, в обращении матрицы коэффициентов $\|a_{ij}\|_{j \in B_1}$) отпадает.

*) Величины Δ_j естественно называть оценками векторов условий задачи (относительно рассматриваемого опорного плана).

Компоненты x_i начального опорного плана также определяются из системы линейных уравнений

$$\sum_{i \in B_1} x_i A_i = B.$$

Числа z_j , формирующие Δ_j , и значение линейной формы вычисляются соответственно по формулам (3.9) и (3.11). Все эти относительно громоздкие вычисления необходимы, однако, только на первом шаге. Как мы сейчас увидим, можно получить рекуррентные формулы, позволяющие по значениям параметров на предыдущем шаге легко вычислять соответствующие параметры на каждом последующем шаге.

Перейдем к выводу рекуррентных формул. Формула (3.17) определяет разложение вектора условий по векторам базиса. Эта формула имеет место для любого вектора условий A_j . В частности, эта формула справедлива для вектора A_k , который целесообразно ввести в базис. В этом случае формуле (3.17) удобно придать вид

$$A_k = \sum_{i \in B_1} z_{ik} A_i = z_{rk} A_r + \sum_{\substack{i \in B_1 \\ i \neq r}} z_{ik} A_i. \quad (3.18)$$

Символ $i \in B_1, i \neq r$ означает здесь, что при суммировании учитываются все индексы i , соответствующие векторам базиса, за исключением индекса r . Под r будем понимать номер вектора, который следует заменить в базисе B_1 при переходе к новому опорному плану.

Последующий базис B_2 отличается от предыдущего базиса B_1 только одним вектором. Вместо вектора $A_r \in B_1$ B_2 содержит вектор A_k . Все остальные векторы у двух последовательных базисов общие. Это означает, что следующие два множества индексов совпадают:

$$\left\{ \begin{array}{l} i \in B_1 \\ i \neq r \end{array} \right\} \equiv \left\{ \begin{array}{l} i \in B_2 \\ i \neq k \end{array} \right\}. \quad (3.19)$$

Полученное соотношение окажется полезным для последующих рассуждений.

Воспользуемся формулой (3.18) для того, чтобы выразить разложение вектора A_r по векторам нового базиса. Имеем

$$A_r = \frac{1}{z_{rk}} A_k - \sum_{\substack{i \in B_1 \\ i \neq r}} \frac{z_{ik}}{z_{rk}} A_i. \quad (3.20)$$

Формула (3.20) позволяет придать иной вид исходному соотношению (3.17)

$$\begin{aligned} A_j &= \sum_{i \in B_1} z_{ij} A_i = \sum_{\substack{i \in B_1 \\ i \neq r}} z_{ij} A_i + z_{rj} A_r = \\ &= \sum_{\substack{i \in B_1 \\ i \neq r}} z_{ij} A_i + \frac{z_{rj}}{z_{rk}} A_k - \sum_{\substack{i \in B_1 \\ i \neq r}} z_{ik} \frac{z_{rj}}{z_{rk}} A_i \end{aligned}$$

или, группируя выражения при одинаковых векторах условий,

$$A_j = \sum_{\substack{i \in B_1 \\ i \neq r}} \left(z_{ij} - z_{ik} \frac{z_{rj}}{z_{rk}} \right) A_i + \frac{z_{rj}}{z_{rk}} A_k. \quad (3.21)$$

С другой стороны, разложение вектора условий A_j по векторам нового базиса B_2 по аналогии с (3.17) может быть записано в виде

$$A_j = \sum_{i \in B_2} z'_{ij} A_i,$$

где z'_{ij} — компоненты разложения A_j по векторам нового базиса — параметры очередного приближения.

Соотношение (3.19) позволяет переписать последнее разложение в виде

$$A_j = \sum_{\substack{i \in B_2 \\ i \neq k}} z'_{ij} A_i + z'_{kj} A_k = \sum_{\substack{i \in B_1 \\ i \neq r}} z'_{ij} A_i + z'_{kj} A_k. \quad (3.22)$$

Но вектор A_j имеет единственное разложение по векторам базиса B_2 . Сравнивая разложения (3.21) и (3.22), получаем рекуррентные формулы, связывающие состав-

ляющие любого вектора условий в двух соседних шагах метода последовательного улучшения плана

$$z'_{ij} = z_{ij} - z_{ik} \frac{z_{rj}}{z_{rk}} \text{ при } i \in B_2, i \neq k, \quad (3.23)$$

$$z'_{kj} = \frac{z_{rj}}{z_{rk}}. \quad (3.24)$$

Формулы (3.23) и (3.24) являются исходными для построения экономного алгоритма вычислений.

3.2. Покажем, что формулы (3.23), (3.24) могут быть использованы также для вычисления других параметров последующих приближений по характеристикам предшествующих итераций. В частности, аналогичные рекуррентные формулы применяются для перехода от одного опорного плана к следующему и от оценок $\Delta_j = z_j - c_j$ предшествующего плана к оценкам $\Delta'_j = z'_j - c_j$ последующего приближения.

Опорный план представляет собой набор коэффициентов разложения вектора ограничений по векторам базиса. Формулы (3.2) и (3.8) при соответствующем подборе θ определяют разложение вектора ограничений по векторам двух последовательных базисов. Перепишем эти формулы в новых обозначениях, не предполагающих специальной нумерации компонент начального опорного плана,

$$B = \sum_{i \in B_1} x_i A_i,$$

$$B = \sum_{i \in B_1} (x_i - \theta z_{ik}) A_i + \theta A_k. \quad (3.25)$$

При этом

$$\theta = \min_i \frac{x_i}{z_{ik}}, \quad z_{ik} > 0. \quad (3.26)$$

(В формулах (3.25) и (3.26) индекс j заменен на k . Так мы обозначаем индекс вектора, вводимого в базис. Обычно индекс k соответствует наименьшему Δ_k .)

Пусть минимум в формуле (3.26) достигается при $i = r$. Это значит, что

$$\theta = \frac{x_r}{z_{rk}}.$$

В этом случае коэффициент при A_r в формуле (3.25) обратится в нуль, и вектор A_k заменит в базисе вектор A_r . Формула (3.25) примет вид

$$B = \sum_{\substack{i \in B_1 \\ i \neq r}} \left(x_i - \frac{x_r}{z_{rk}} z_{ik} \right) A_i + \frac{x_r}{z_{rk}} A_k.$$

С другой стороны, обозначая через x_i' компоненты опорного плана, соответствующего новому базису B_2 , имеем

$$B = \sum_{i \in B_2} x_i' A_i.$$

Рассуждая так же, как и при получении формул (3.23), (3.24), получим

$$x_i' = x_i - \frac{x_r}{z_{rk}} z_{ik}, \quad i \in B_2, \quad (i \neq k),$$

$$x_k' = \frac{x_r}{z_{rk}}.$$

Аналогия с формулами (3.23), (3.24) будет полная, если изменить обозначения вектора ограничений и его составляющих: $B = A_0$, $x_i = z_{i0}$ и соответственно $x_i' = z_{i0}'$. В этих обозначениях формулы (3.23), (3.24) могут быть использованы для вычисления последовательных составляющих векторов условий и вектора ограничений.

3.3. Нетрудно убедиться в том, что эти же рекуррентные формулы могут быть применены для вычисления оценок Δ_i' последующего приближения по оценкам Δ_i предыдущего. Перепишем формулу (3.9) в принятых в настоящем параграфе обозначениях

$$z_j = \sum_{i \in B_1} c_i z_{ij}. \quad (3.27)$$

Соответственно для z_j' имеем

$$z_j' = \sum_{i \in B_2} c_i z_{ij}'.$$

Следующие простые преобразования приводят нас к требуемому выводу

$$z_j' = \sum_{i \in B_2} c_i z_{ij}' = \sum_{\substack{i \in B_2 \\ i \neq k}} c_i z_{ij}' + c_k z_{kj}' = \sum_{\substack{i \in B_1 \\ i \neq r}} c_i z_{ij}' + c_k z_{kj}'.$$

Воспользуемся рекуррентными формулами (3.23), (3.24). Получим

$$z_j' = \sum_{\substack{i \in B_1 \\ i \neq r}} c_i \left(z_{ij} - z_{ik} \frac{z_{rj}}{z_{rk}} \right) + c_k \frac{z_{rj}}{z_{rk}}.$$

Символ $i \neq r$ под знаком суммы можно изъять за ненадобностью, так как при $i = r$ соответствующее слагаемое обращается в нуль. Поэтому

$$\begin{aligned} z_j' &= \sum_{i \in B_1} c_i \left(z_{ij} - z_{ik} \frac{z_{rj}}{z_{rk}} \right) + c_k \frac{z_{rj}}{z_{rk}} = \\ &= \sum_{i \in B_1} c_i z_{ij} - \frac{z_{rj}}{z_{rk}} \sum_{i \in B_1} c_i z_{ik} + c_k \frac{z_{rj}}{z_{rk}}. \end{aligned}$$

Формула (3.27) позволяет переписать последнее соотношение в виде

$$z_j' = z_j - \frac{z_{rj}}{z_{rk}} (z_k - c_k).$$

Вычтем из обеих частей равенства c_j . Учитывая принятые обозначения, получим

$$\Delta_j' = \Delta_j - \frac{z_{rj}}{z_{rk}} \Delta_k.$$

Чтобы сохранить аналогию с формулой (3.23), обозначим

$$\Delta_j = z_{0j}.$$

Обозначим, кроме того, значение линейной формы через z_{00}

$$z_{00} = L(X).$$

Это обозначение оправдывается тем, что вычисление величины линейной формы, соответствующей новому опорному плану, по ее значению на предыдущем шаге производится по той же рекуррентной формуле (3.23). Действительно, в соответствии с формулой (3.11), связывающей значения линейной формы для двух последовательных опорных планов, имеем

$$L(X') = L(X) - \theta \Delta_k.$$

Здесь индекс j заменен на индекс k , чтобы подчеркнуть, что в базис вводится вектор A_k .

Согласно принятым в этом параграфе обозначениям, последняя формула может быть переписана в виде

$$z'_{00} = z_{00} - \frac{z_{r0}}{z_{rk}} z_{0k}.$$

3.4. Мы пришли к следующему выводу.
Рекуррентные формулы

$$z'_{ij} = z_{ij} - \frac{z_{rj}}{z_{rk}} z_{ik} \text{ (для } i \neq k),$$

$$z'_{kj} = \frac{z_{rj}}{z_{rk}}$$

могут быть использованы для перехода от одного плана к другому. Они позволяют вычислять следующие параметры каждого шага метода последовательного улучшения плана (если известны соответствующие параметры предыдущего приближения):

- а) компоненты разложения всех векторов условий по векторам базиса ($i \in B_1, j = 1, 2, \dots, n$);
- б) составляющие опорного плана ($i \in B_1, j = 0$);
- в) оценки Δ_j ($i = 0, j = 1, 2, \dots, n$);
- г) значения линейной формы ($i = 0, j = 0$).

§ 4. I АЛГОРИТМ МЕТОДА ПОСЛЕДОВАТЕЛЬНОГО УЛУЧШЕНИЯ ПЛАНА

4.1. Полученные в предыдущем параграфе рекуррентные формулы сводят решение задачи линейного программирования к составлению и анализу последовательности таблиц. Все элементы каждой последующей таблицы весьма просто вычисляются через элементы предыдущей таблицы. С некоторыми трудностями связано только составление первой таблицы.

Пусть для определенности начальный базис составлен из векторов условий, имеющих m первых номеров $B_1 = \{A_1, A_2, \dots, A_m\}$.

Табл. I представляет собой исходный этап вычислений.

Таблица I

c_l	A_j										c_k		c_n	
	c_1	c_2	c_r	c_{m-1}	c_m	c_{m+1}	c_j	c_k	c_n	c_n	c_n	c_n	c_n	c_n
B_1	$B=A_0$	A_1	A_2	A_r	A_{m-1}	A_m	A_{m+1}	A_j	A_k	A_n	A_n	A_n	A_n	A_n
c_1	A_1	z_{10}	1	$\dots$	$\dots$	$\dots$	$z_{1,m+1}$	z_{1j}	z_{1k}	z_{1n}	z_{1k}	z_{1k}	z_{1n}	z_{1n}
c_2	A_2	z_{20}	1	$\dots$	$\dots$	$\dots$	$z_{2,m+1}$	z_{2j}	z_{2k}	z_{2n}	z_{2k}	z_{2k}	z_{2n}	z_{2n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_l	A_l	z_{l0}	$\dots$	$\dots$	$\dots$	$\dots$	$z_{l,m+1}$	$z_{lj} \text{ (I)}$	$z_{lk} \text{ (II)}$	z_{ln}	$z_{lk} \text{ (II)}$	$z_{lk} \text{ (II)}$	z_{ln}	z_{ln}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_r	A_r	z_{r0}	1	$\dots$	$\dots$	$\dots$	$z_{r,m+1}$	$z_{rj} \text{ (III)}$	$z_{rk} \text{ (IV)}$	z_{rn}	$z_{rk} \text{ (IV)}$	$z_{rk} \text{ (IV)}$	z_{rn}	z_{rn}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_m	A_m	z_{m0}	$\dots$	$\dots$	1	$\dots$	$z_{m,m+1}$	z_{mj}	z_{mk}	z_{mn}	z_{mk}	z_{mk}	z_{mn}	z_{mn}
z_j	z_0	z_1	z_2	z_r	z_{m-1}	z_m	z_{m+1}	z_j	z_k	z_n	z_k	z_k	z_n	z_n
$\Delta_j = z_j - c_j$	z_{00}	z_{01}	z_{02}	z_{0r}	$z_{0,m-1}$	z_{0m}	$z_{0,m+1}$	z_{0j}	z_{0k}	z_{0n}	z_{0k}	z_{0k}	z_{0n}	z_{0n}

Таблица содержит коэффициенты разложения z_{ij} вектора ограничений $B = A_0$ и векторов условий A_j по векторам базиса.

Первый столбец таблицы — разложение вектора ограничений по векторам базиса. Эти компоненты представляют собой исходный опорный план. Ненулевые переменные x_i , входящие в опорный план, обозначены здесь через z_{i0} . Следующие m столбцов соответствуют векторам базиса. Ясно, что каждый из них имеет одну составляющую, равную единице, а остальные — нули. Нули в таблице не пишутся. Несмотря на тривиальный характер элементов этих столбцов, они не исключаются из таблицы. Этого требует удобство последующих вычислений.

Остальные столбцы определяют компоненты разложения z_{ij} векторов условий, не вошедших в базис, по соответствующим векторам базиса. В общем случае величины $z_{ij} (i = 1, 2, \dots, m)$ вычисляются из систем уравнений

$$A_j = \sum_{i \in B_1} z_{ij} A_i.$$

Здесь m уравнений с m неизвестными $z_{1j}, z_{2j}, \dots, z_{mj}$

$$a_{11}z_{1j} + a_{12}z_{2j} + \dots + a_{1m}z_{mj} = a_{1j},$$

$$\dots \dots \dots$$

$$a_{m1}z_{1j} + a_{m2}z_{2j} + \dots + a_{mm}z_{mj} = a_{mj}.$$

Система уравнений имеет единственное решение, поскольку исходный план является опорным планом.

При большом количестве условий в задаче линейного программирования вычисление компонент z_{ij} оказывается наиболее трудоемкой частью работы. В ряде случаев увеличение числа переменных (использование так называемого M -метода) может сократить количество вычислений. В M -методе начальный базис соответствует дополнительным переменным. Каждый из дополнительных векторов условий имеет одну единичную компоненту, а остальные — нулевые. Разложение векторов условий по такому базису не представляет труда. Коэффициент

разложения j -го вектора условий A_j при i -м векторе базиса совпадает с соответствующей составляющей вектора A_j (§ 2)

$$z_{ij} = a_{ij}.$$

По той же причине при базисе, составленном из единичных векторов,

$$x_i = z_{i0} = b_i.$$

В этом случае составление исходной таблицы не требует никаких вычислений. Столбцы B и A_j табл. I заполняются непосредственно заданными значениями компонент b_i и a_{ij} .

Левый столбец табл. I содержит значения коэффициентов линейной формы (c_i), соответствующих компонентам исходного опорного плана. Если исходный базис составлен из дополнительных векторов, то левый столбец не заполняется. (Дополнительным переменным соответствуют нулевые коэффициенты в линейной форме.) Столбец c_i необходим для вычисления z_j — элементов предпоследней строки табл. I. Величины z_j вычисляются по формуле

$$z_j = \sum_{i \in B_1} c_i z_{ij}.$$

Первый элемент предпоследней строки, как нетрудно сообразить, равен значению линейной формы для исходного опорного плана

$$z_0 = \sum_{i \in B_1} c_i z_{i0} = \sum_{i \in B_1} c_i x_i = L(X).$$

Очевидно, что элементы предпоследней строки, отвечающие векторам базиса, совпадают с соответствующими коэффициентами c_j линейной формы. Последняя строка табл. I заполняется значениями $\Delta_j = z_j - c_j$. Ясно, что $\Delta_j = 0$ при $j \in B_1$. Для удобства вычислений в верхней строке таблицы над наименованиями векторов условий проставлены соответствующие значения коэффициентов линейной формы c_j .

Как уже указывалось, для преобразования всех элементов таблицы по единым рекуррентным формулам

целесообразно обозначать элементы последней строки через z_{0j} . Ясно, что элемент в левом нижнем углу таблицы равен

$$z_{00} = z_0 - c_0 = z_0 = L(X).$$

Элементы последней строки таблицы представляют собой оценки Δ_j векторов условий A_j по отношению к выбранному опорному плану. В соответствии с критерием оптимальности исходный план обращает линейную форму в максимум, если все элементы последней строки таблицы неотрицательны. Если имеет место такой случай, то все вычисления ограничиваются составлением первой таблицы.

Пусть теперь среди элементов последней строки окажутся отрицательные величины. Прежде чем переходить к составлению следующей таблицы, необходимо установить, нет ли оснований считать линейную форму неограниченной в области своего определения. Ранее было показано, что максимум линейной формы бесконечен, если все компоненты z_{ij} вектора условий A_j , для которого $\Delta_j < 0$, оказываются отрицательными. Если неограниченность линейной формы установлена, то решение задачи закончено.

Пусть теперь все векторы условий, для которых $\Delta_j < 0$, имеют положительные составляющие. Необходимо выбрать такой вектор A_k , который должен заменить один из векторов базиса и увеличить при этом значение линейной формы. В качестве вектора A_k можно, вообще говоря, принять любой из векторов условий, для которого оценка Δ отрицательна. Обычно (но не всегда) число шагов уменьшается, если в базис вводится вектор, для которого Δ наименьшее (т. е. наибольшее по абсолютной величине). Затем следует определить вектор A_r , исходного базиса, который должен быть изъят из базиса и заменен на вектор A_k .

Выбор A_r осуществляется следующим образом. Выделяются положительные компоненты A_k и составляются отношения соответствующих компонент вектора ограничений (столбец $A_0 = B$) к положительным составляющим A_k . Номер вектора, который должен быть изъят из

базиса, равен значению i , при котором достигается минимум отношения

$$\theta = \min_{\substack{i \\ z_{ik} > 0}} \frac{x_i}{z_{ik}} = \min_{\substack{l \\ z_{lk} > 0}} \frac{z_{l0}}{z_{lk}}.$$

(Отношения $\frac{z_{l0}}{z_{lk}}$, для которых $z_{lk} > 0$, целесообразно выписывать в отдельный столбец справа от таблицы против соответствующих строк.) В невырожденной задаче линейного программирования минимум достигается для одного значения i . Выбрав векторы A_k и A_r , можно перейти к составлению табл. II. Форма табл. II не отличается от формы первой таблицы. В ней можно только исключить промежуточную строку z_j . Однако следует учесть, что величины z_j легко вычисляются двумя способами. С одной стороны, z_j можно вычислить непосредственно

$$z_j = \sum_{i \in B} c_i z_{ij}.$$

С другой стороны, z_j можно получить по вычисленной при помощи рекуррентной формулы оценке Δ_j

$$\Delta_j = z_j - c_j.$$

Создается возможность контролировать вычисления. Поэтому не следует рекомендовать исключение строки z_j из второй и последующих таблиц, несмотря на то, что в непосредственных вычислениях величины z_j , начиная со второго шага, не участвуют.

В табл. II в столбец B_2 вносятся все векторы старого базиса, за исключением A_r . Вместо вектора A_r вводится новый вектор A_k . Соответственно в той же строке слева вместо c_r записывается величина c_k . Порядок столбцов в табл. II сохраняется таким же, как и в первой таблице.

Все элементы табл. II вычисляются в соответствии с рекуррентными формулами (3.23) и (3.24) по данным первой таблицы. Вначале заполняется строка, отвечающая вновь введенному в базис вектору. Элементы этой строки z'_{kj} вычисляются по формуле

$$z'_{kj} = \frac{z_{rj}}{z_{rk}}, \quad j = 1, 2, \dots, n,$$

Таблица II

c_l		c_1	c_2	c_r	c_m	c_{m+1}	c_j	c_k	c_n
	B_2								
		A_0	A_2	A_r	A_m	A_{m+1}	A_j	A_k	A_n
c_1	A_1	z'_{10}	1	$z'_{1,r}$	$z'_{1,m+1}$	$z'_{1,m+1}$	z'_{ij}	z'_{1n}	z'_{1n}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_{r-1}	A_{r-1}	$z'_{r-1,0}$	$z'_{r-1,r}$	$z'_{r-1,m+1}$	$z'_{r-1,j}$	$z'_{r-1,n}$	$z'_{r-1,n}$	$z'_{r-1,n}$	$z'_{r-1,n}$
c_k	A_k	z'_{k0}	z'_{kr}	$z'_{k,m+1}$	z'_{kj}	z'_{kn}	1	z'_{kn}	z'_{kn}
c_{r+1}	A_{r+1}	$z'_{r+1,0}$	$z'_{r+1,r}$	$z'_{r+1,m+1}$	$z'_{r+1,j}$	$z'_{r+1,n}$	$z'_{r+1,n}$	$z'_{r+1,n}$	$z'_{r+1,n}$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
c_m	A_m	z'_{m0}	z'_{mr}	1	$z'_{m,m+1}$	z'_{mn}	z'_{mj}	z'_{mn}	z'_{mn}
z_j		z'_0	z'_1	z'_2	z'_m	z'_{m+1}	z'_j	z'_k	z'_n
Δ_j		z'_{00}	z'_{01}	z'_{02}	z'_{0m}	$z'_{0,m+1}$	z'_{0j}	z'_{0k}	z'_{0n}

где z_{rk} — элемент, записанный в табл. I на пересечении столбца A_k — вектора, который вводится в базис, и строки A_r , отвечающей вектору, который выводится из базиса; z_{rj} ($j = 0, 1, \dots, n$) — элементы строки A_r табл. I.

Нетрудно также заполнить столбец A_k табл. II. k -й вектор условий стал вектором базиса B_2 , заменив вектор A_r в B_1 . Поэтому столбец A_k табл. II содержит единицу в r -й строке, c_k в $(m+1)$ -й строке и нули во всех остальных.

Рекуррентная формула

$$z'_{ij} = z_{ij} - \frac{z_{rj}z_{ik}}{z_{rk}} \quad (3.23)$$

приводит к следующему, не очень экономному, но удобному для запоминания правилу заполнения остальных клеток табл. II.

Для вычисления i -го элемента j -го столбца z'_{ij} следует представить себе в табл. I прямоугольник с вершинами в точках: z_{ij} (I), z_{ik} (II), z_{rj} (III) и z_{rk} (IV).

Значение z'_{ij} получится, если из элемента, соответствующего I вершине, вычесть произведение элементов, отвечающих II и III вершинам, деленное на число, расположенное в IV вершине прямоугольника.

В первой таблице целесообразно выделять (обводить рамкой) k -й столбец (вектор, вводимый в базис) и r -ю строку (отвечающую вектору, исключаемому из базиса). При ручном счете это облегчает пользование «правилом прямоугольника».

Несколько менее удобно, но более экономно правило, основанное на следующей записи рекуррентной формулы, учитывающей как соотношение (3.23), так и формулу (3.24),

$$z'_{ij} = z_{ij} - z'_{kj}z_{ik}.$$

Пользуясь любым из приведенных правил, заполняем всю табл. II, в том числе и первый столбец, соответствующий новому опорному плану, и последнюю строку, содержащую новые значения оценок Δ'_j . Если не все Δ'_j отрицательны, следует приступить к составлению новой таблицы. Следующая таблица составляется по табл. II,

исходя из тех же правил, по каким табл. II получена из табл. I.

В невырожденной задаче линейного программирования значение линейной формы (элемент z_{00} , стоящий в левом нижнем углу) при переходе от таблицы к таблице возрастает. Вычисления прекращаются, когда все Δ_j оказываются неотрицательными и, следовательно, достигается максимум линейной формы, или когда устанавливается неограниченность линейной формы.

Как видим, при наличии исходного опорного плана метод последовательного улучшения плана связан с простыми вычислениями. Приведенный алгоритм весьма удобен для использования на универсальных цифровых вычислительных машинах. Однако количество шагов (однотипных вычислений) при большом числе переменных и условий может оказаться достаточно большим.

4.2. Рассмотрим два примера.

Пример 1. Требуется обратить в максимум линейную форму

$$L = -2x_1 - 3x_2 + 2x_4 + 3x_5$$

при условиях:

$$\begin{aligned} 2x_1 - x_2 + x_3 &\leq 12, \\ x_1 + x_4 &\leq 5, \\ 2x_1 + 2x_2 + x_3 - 2x_5 &\leq 20, \\ x_1 - x_2 - 2x_3 + 2x_4 - 2x_5 &\leq 10, \\ -2x_1 + 2x_2 - 2x_3 - 2x_4 + x_5 &\leq 24, \\ x_i &\geq 0, \quad i = 1, 2, 3, 4, 5. \end{aligned}$$

Введем дополнительные неотрицательные переменные и преобразуем условия в равенства. Имеем

$$\begin{aligned} 2x_1 - x_2 + x_3 + x_6 &= 12, \\ x_1 + x_4 + x_7 &= 5, \\ 2x_1 + 2x_2 + x_3 - 2x_5 + x_8 &= 20, \\ x_1 - x_2 - 2x_3 + 2x_4 - 2x_5 + x_9 &= 10, \\ -2x_1 + 2x_2 - 2x_3 - 2x_4 + x_5 + x_{10} &= 24, \\ x_i &\geq 0, \quad i = 1, 2, \dots, 10. \end{aligned}$$

Ясно, что $X = \{0, 0, 0, 0, 0, 12, 5, 20, 10, 24\}$ является опорным планом. Следовательно, начальный базис состоит из векторов $A_6, A_7, A_8, A_9, A_{10}$. Векторы исходного базиса образуют единичную матрицу. Поэтому компо-

ненты разложения всех векторов условий и вектора ограничений по векторам базиса совпадают с соответствующими составляющими этих векторов.

Составим табл. I.

Таблица I

c_i								-2	-3		2	3
	B_1	A_j										
		$A_0=B$	A_6	A_7	A_8	A_9	A_{10}	A_1	A_2	A_3	A_4	A_5
	A_6	12	1					2	-1	1		
	A_7	5		1				1			1	
	A_8	20			1			2	2	1		-2
	A_9	10				1		1	-1	-2	2	-2
	A_{10}	24					1	-2	2	-2	-2	1
z_j												
Δ_j								2	3		-2	-3

Векторы исходного базиса отвечают дополнительным переменным, которые входят в линейную форму с нулевыми коэффициентами. Поэтому левый столбец таблицы не заполнен, все $z_j = \sum_{i \in B_1} c_i z_{ij}$ равны нулю и, следовательно, $\Delta_j = -c_j$. Значение линейной формы $z_{00} = 0$.

В столбцах с отрицательными значениями Δ_j не все составляющие неотрицательны. В базис B_2 следует ввести A_5 , так как ему соответствует наименьшее значение Δ_j . Среди компонент A_5 только одно положительное число. Это — составляющая разложения A_5 при векторе A_{10} из базиса B_1 ($z_{10,5}$). Поэтому исключать из базиса следует вектор A_{10} .

Составляем табл. II. В табл. II место вектора A_{10} займет вектор A_5 . Заполним строку, соответствующую вновь введенному в базис вектору A_5 . Элементы этой строки получаются из элементов строки A_{10} табл. I делением их на $z_{10,5}$. В нашем случае $z_{10,5} = 1$, и строка A_5 табл. II

не будет отличаться от строки A_{10} табл. I. Поскольку вектор A_5 вводится в базис, все его компоненты в табл. II, за исключением последней, равны нулю, а $z'_{55} = 1$. Пользуясь правилом

$$z'_{ij} = z_{ij} - z'_{5j} \cdot z_{i5}$$

(где z'_{5j} — элементы только что заполненной строки табл. II, а z_{i5} — компоненты столбца A_5 из табл. I), заполним всю табл. II.

Имеем, например,

$$\begin{aligned} z'_{6,0} &= z_{6,0} - z'_{5,0} \cdot z_{6,5} = 12 - 24 \cdot 0 = 12, \\ z'_{8,0} &= z_{8,0} - z'_{5,0} \cdot z_{8,5} = 20 - 24(-2) = 68, \\ z'_{7,4} &= z_{7,4} - z'_{5,4} \cdot z_{7,5} = 1 - (-2) \cdot 0 = 1, \\ z'_{9,10} &= z_{9,10} - z'_{5,10} \cdot z_{9,5} = 0 - 1 \cdot (-2) = 2, \\ z'_{00} &= z_{00} - z'_{5,0} \cdot z_{0,5} = 0 - 24 \cdot (-3) = 72. \end{aligned}$$

Таблица II

c_l								-2	-3		2	3
	B_2	A_j										
		$B = A_0$	A_6	A_7	A_8	A_9	A_{10}	A_1	A_2	A_3	A_4	A_5
	A_6	12	1					2	-1	1		
←	A_7	5		1				1			1	
	A_8	68			1		2	-2	6	-3	-4	
	A_9	58				1	2	-3	3	-6	-2	
→	3 A_5	24					1	-2	2	-2	-2	1
	z_j	72					3	-6	6	-6	-6	3
	Δ_j	72					3	-4	9	-6	-8	

В табл. II из всех векторов базиса B_2 только вектору A_5 соответствует коэффициент линейной формы, отличный от нуля ($c_5 = 3$).

Для контроля вычислений можно получить значения линейной формы и Δ_j не только по рекуррентным формулам, но и непосредственно

$$\Delta_j := z_j - c_j = \sum_{i \in B} c_i z_{ij} - c_j.$$

Например,

$$\Delta_2' = 0 \cdot (-1) + 0 \cdot 0 + 0 \cdot 6 + 0 \cdot 3 + 3 \cdot 2 - (-3) = 9.$$

Тот же результат получаем по рекуррентной формуле

$$\Delta_2' = z_{02}' = 3 - 2 \cdot (-3) = 9.$$

Значение линейной формы в табл. II возросло до 72.

Переходим к составлению табл. III. Сравнивая элементы последней строки табл. II, убеждаемся в целесообразности ввода в новый базис вектора A_4 . Этот вектор имеет единственную положительную компоненту $z_{7,4}$. Это значит, что вектор A_4 войдет в базис вместо A_7 .

Последующие вычисления ничем не отличаются от расчетов, проведенных при переходе от табл. I к табл. II.

Т а б л и ц а III

c_i									-2	-3		2	3
	B_3	A_j											
		$B = A_0$	A_6	A_7	A_8	A_9	A_{10}	A_1	A_2	A_3	A_4	A_5	
←	A_6	12	1					2	-1	1			
→	2 A_4	5		1				1			1		
	A_8	88		4	1		2	2	6	-3			
	A_9	68		2		1	2	-1	3	-6			
	3 A_5	34		2			1		2	-2		1	
	z_j	112		8			3	2	6	-6	2	3	
	Δ_j	112		8			3	4	9	-6			

Таким же образом составляется табл. IV.

Таблица IV

c_l									-2	-3		2	3
	B_i	A_j											
		$B = A_0$	A_6	A_7	A_8	A_9	A_{10}	A_1	A_2	A_3	A_4	A_5	
→	A_3	12	1					2	-1	1			
2	A_4	5		1				1			1		
	A_8	124	3	4	1		2	8	3				
	A_9	140	6	2		1	2	11	-3				
3	A_5	58	2	2			1	4					1
z_j		184	6	8			3	14			2	3	
Δ_j		184	6	8			3	16	3				

В табл. IV в базис введен A_3 вместо A_6 . Во всех таблицах стрелками указаны векторы условий, вводимые в базис и исключаемые из базиса на соответствующих этапах.

В табл. IV все разности Δ_j оказались неотрицательными. Следовательно, опорный план, отвечающий этой таблице, является решением задачи линейного программирования.

Итак,

$$X_{\text{опт}} = (0, 0, 12, 5, 58, 0, 0, 124, 140, 0);$$

$$L(X_{\text{опт}}) = 184.$$

В табл. V записаны все этапы решения примера 2. Требуется вычислить максимум линейной формы

$$L = -2x_1 - 3x_2 + 2x_4 + 3x_5 - x_7 - x_8 + \\ + 2x_9 - x_{10} - 3x_{11} + 2x_{12} - x_{13} + x_{14} + 2x_{15}$$

при условиях:

$$\begin{aligned}
 &2x_1 - x_2 + x_3 + x_6 + x_7 + x_9 - x_{10} - 2x_{11} - x_{12} + \\
 &\quad + 2x_{13} + x_{15} \leq 12, \\
 &x_1 + x_4 - x_6 + x_7 + 2x_8 + x_{10} + x_{11} + x_{12} + x_{14} \leq 5, \\
 &2x_1 + 2x_2 + x_3 - 2x_5 + x_6 + 2x_7 - x_8 + \\
 &\quad + x_9 - 2x_{10} + 2x_{12} - 2x_{13} + 2x_{15} \leq 20, \\
 &x_1 - x_2 - 2x_3 + 2x_4 - 2x_5 + x_6 - x_8 + \\
 &\quad + 2x_{10} - 2x_{11} - x_{12} + 2x_{13} \leq 10, \\
 &-2x_1 + 2x_2 - 2x_3 - 2x_4 + x_5 + x_6 + x_7 + \\
 &\quad + x_8 + x_{10} + 2x_{11} + 2x_{12} + x_{14} - 2x_{15} \leq 24, \\
 &x_i \geq 0, \quad i = 1, 2, \dots, 15.
 \end{aligned}$$

Приведем задачу к канонической форме. Для этого введем дополнительные неотрицательные переменные по одному в каждое условие. Эти переменные определяют исходный опорный план и начальный базис. Табл. V определяет последовательность вычислений и специальных пояснений не требует.

Итак, оптимальный план, отвечающий рассматриваемой задаче линейного программирования, имеет следующие 5 ненулевых компонент

$$x_4 = 5, \quad x_5 = 58, \quad x_{15} = 12, \quad x_{18} = 112, \quad x_{19} = 116.$$

Максимальное значение линейной формы равно 208.

В приведенных примерах не было необходимости выписывать отношения $\frac{z_{i0}}{z_{ik}}$ для $z_{ik} > 0$ в отдельный столбец справа от таблицы, поскольку в каждом из рассмотренных случаев оказывалось лишь одно положительное z_{ik} .

Рассмотрим теперь простой пример, в котором по ходу решения задачи устанавливается ее неразрешимость.

Пусть требуется вычислить максимум линейной формы

$$L = x_1 + 2x_2 + 2x_3 + x_4$$

при условиях

$$x_1 - x_3 + \frac{1}{2}x_4 = 1,$$

$$x_2 + x_3 - x_4 = 1,$$

$$x_i \geq 0, \quad i = 1, 2, 3, 4.$$

[illegible]

	A_{18}	68	1	2	-2	6	-3	-4	3	4	1	1	4	6	-2	2	-2
	A_{19}	58	1	2	-3	3	-6	-2	3	2	1	4	2	3	2	2	-4
3	A_3	24		1	-2	2	-2	-2	1	1	1	1	2	2	1	1	-2
Δ_j		72		3	-4	9	-6	-8	3	4	4	-2	9	4	1	2	-8
	A_{16}	12	1	2	-1	1	1	1	1	1	1	-1	-2	-1	2	1	1
2	A_4	5	1	1	1	1	1	1	-1	1	2	1	1	1	1	1	1
	A_{18}	88	4	1	2	2	6	-3	-1	8	9	1	4	8	10	-2	6
	A_{19}	68	2	1	2	-1	3	-6	1	4	5	6	4	5	2	4	-4
3	A_5	34	2	1	2	-2	2	-2	1	-1	3	5	3	4	4	3	-2
Δ_j		112	8	3	4	9	-6		-5	12	20	-2	12	17	12	1	10

→

←

→

[illegible]

Исходный опорный план здесь равен $X = (1, 1, 0, 0)$. Следовательно, начальный базис состоит из векторов A_1 и A_2 .

Табл. I для этого примера имеет следующий вид.

Т а б л и ц а I

c_l			1	2	2	1
	B_1	A_j				
		$A_0 = B$	A_1	A_2	A_3	A_4
1	A_1	1	1		-1	$1/2$
2	A_2	1		1	1	-1
Δ_j		3			-1	$-5/2$

В столбцах с отрицательными значениями Δ_j не все составляющие отрицательные. В базис B_2 следует ввести вектор A_4 (ему соответствует наименьшее значение Δ_j). Среди компонент A_4 только одно положительное число $z_{1,4} = \frac{1}{2}$. Поэтому исключать из базиса следует A_1 . В табл. II место вектора A_1 займет A_4 .

Т а б л и ц а II

c_l			1	2	2	1
	B_2	A_j				
		$A_0 = B$	A_1	A_2	A_3	A_4
1	A_4	2	2		-2	1
2	$-A_2$	3	2	1	-1	
Δ_j		8	5		-6	

Здесь $\Delta_3 = -6 < 0$, и обе компоненты разложения вектора A_3 по векторам базиса B_2 отрицательны. В соответствии со сформулированными ранее правилами это означает, что линейная форма задачи неограниченна. В этом легко также убедиться непосредственно, если вычислить x_4 и x_2 из условий и подставить их выражения в линейную форму L .

До сих пор речь шла о задачах линейного программирования, в которых требовалось вычислить максимум

линейной формы. Порядок расчетов остается таким же и при решении задач на минимум. Единственное отличие — в критерии оптимальности и связанном с ним выборе вектора A_k , который следует вводить в базис. В критерии оптимальности знак неравенства изменяется на обратный. Линейная форма достигает минимума, если все Δ_j оказываются неположительными. В базис следует вводить вектор A_k , которому соответствует максимальное положительное значение Δ_k . Все остальные операции проводятся по тем же правилам, по которым определяется максимум линейной формы.

§ 5. ВТОРАЯ ФОРМА КРИТЕРИЯ ОПТИМАЛЬНОСТИ

5.1. Связи между параметрами задачи линейного программирования позволяют получить еще один вычислительный алгоритм метода последовательного улучшения плана. Из самого построения II алгоритма станет ясно, в каких случаях его следует предпочесть первому.

Метод последовательного улучшения плана в форме, близкой к излагаемой в настоящем параграфе, был впервые применен Л. В. Канторовичем к одной из частных задач линейного программирования [11], [8]. (Работа Данцига [28] относится к 1949 г.) Позже (1951 г.) этот алгоритм как одна из реализаций метода разрешающих множителей был использован для решения общей задачи линейного программирования [12].

Второй алгоритм основан на другой записи критерия оптимальности опорного плана. Приведем связанные с этим соображения. Для упрощения записи примем, как и прежде, что исходный базис включает векторы условий $A_1, A_2, \dots, A_m$.

Рассмотрим систему единичных векторов e_i ($i = 1, 2, \dots, m$), i -я компонента e_i равна единице, остальные $m - 1$ составляющие — нули. Все векторы условий по определению могут быть записаны в виде

$$A_j = \sum_{s=1}^m a_{sj} e_s, \quad j = 1, 2, \dots, n. \quad (3.28)$$

Разложим единичные векторы e_s по векторам базиса

$$e_s = \sum_{i=1}^m e_{is} A_i, \quad s = 1, 2, \dots, m. \quad (3.29)$$

Читатель, знакомый с понятием обратной матрицы, без труда обнаружит, что матрицы $\|e_{ij}\|$ и $\|a_{ij}\|$ — взаимно обратные матрицы

$$\|e_{ij}\| = \|a_{ij}\|^{-1}. \quad (3.30)$$

Здесь через $\|a_{ij}\|$ обозначена матрица компонент векторов условий, составляющих рассматриваемый базис*).

Используя равенства (3.28) и (3.29), получаем

$$A_j = \sum_{s=1}^m a_{sj} e_s = \sum_{s=1}^m a_{sj} \left(\sum_{i=1}^m e_{is} A_i \right)$$

или, изменив порядок суммирования,

$$A_j = \sum_{i=1}^m A_i \left(\sum_{s=1}^m a_{sj} e_{is} \right).$$

Сравним полученную формулу с соотношением (3.7).

В силу единственности разложения A_j по векторам базиса имеем

$$z_{ij} = \sum_{s=1}^m a_{sj} e_{is}; \quad i = 1, \dots, m, \quad j = 1, \dots, n. \quad (3.31)$$

Введем m -мерный вектор $\mu = (\mu_1, \mu_2, \dots, \mu_m)$, компоненты которого вычисляются по формуле

$$\mu_k = \sum_{i=1}^m c_i e_{ik}, \quad k = 1, 2, \dots, m. \quad (3.32)$$

Параметры z_j , необходимые для проверки оптимальности опорного плана, связаны простыми соотношениями с компонентами вектора μ и составляющими соответствующих векторов условий. Согласно формулам (3.9), (3.31) и (3.32) имеем

$$z_j = \sum_{i=1}^m c_i z_{ij} = \sum_{i=1}^m c_i \sum_{s=1}^m a_{sj} e_{is} = \sum_{s=1}^m \left(\sum_{i=1}^m c_i e_{is} \right) a_{sj} = \sum_{s=1}^m \mu_s a_{sj}.$$

Критерий оптимальности опорного плана (§ 1) может теперь быть сформулирован следующим образом.

*) Для вычисления $\|e_{ij}\|$ (для обращения матрицы $\|a_{ij}\|$) в случае, когда базис начального плана не является искусственным, можно воспользоваться одним из численных методов линейной алгебры [17].

Опорный план обращает линейную форму в максимум, если

$$\sum_{i=1}^m \mu_i a_{ij} \geq c_j.$$

для всех j от 1 до n . И, соответственно, опорный план обращает линейную форму в минимум, если

$$\sum_{i=1}^m \mu_i a_{ij} \leq c_j$$

для всех j от 1 до n . Все последующие рассуждения мы будем, как и прежде, вести относительно задач на максимум линейной формы.

5.2. Приведенные в этом параграфе соотношения намечают новый порядок вычислений при решении задач линейного программирования по методу последовательного улучшения плана.

На каждом шаге последовательного улучшения плана при заданном исходном опорном плане необходимо выполнить следующие операции.

1°. Вычислить коэффициенты e_{ij} разложения единичных векторов e_i по векторам базиса или, что то же самое, построить матрицу, обратную матрице векторов базиса.

2°. Построить вектор μ с компонентами

$$\mu_k = \sum_{i=1}^m c_i e_{ik}, \quad k = 1, 2, \dots, m.$$

3°. Вычислить параметры z_j

$$z_j = \sum_{i=1}^m \mu_i a_{ij}, \quad j = 1, 2, \dots, n.$$

4°. Установить по критерию оптимальности, является ли выбранный план оптимальным.

Если рассматриваемый опорный план не является решением задачи линейного программирования, то те же правила, что и в I алгоритме, позволяют выяснить, какой вектор условий следует ввести в базис и какой вектор необходимо исключить из базиса для того, чтобы значение линейной формы увеличилось. Последующие шаги алгоритма содержат те же операции. Относи-

тельно громоздкие вычисления связаны только с первым шагом. Переход от начальной таблицы к последующим происходит по рекуррентным формулам, подобным соотношениям (3.23), (3.24).

5.3. Выведем формулы, связывающие значения e_{ij} и μ_j с соответствующими значениями на предыдущем шаге.

Перепишем формулу (3.29), определяющую разложение единичного вектора e_j по векторам базиса B_1 в следующем виде:

$$e_j = \sum_{i \in B_1} e_{ij} A_i.$$

Те же единичные векторы могут быть разложены по новому базису B_2

$$e_j = \sum_{i \in B_2} e'_{ij} A_i.$$

Базис B_2 образуется из предыдущего базиса B_1 заменой вектора A_r на A_k . Поэтому

$$e_j = \sum_{\substack{i \in B_1 \\ i \neq r}} e_{ij} A_i + e_{rj} A_r = \sum_{\substack{i \in B_2 \\ i \neq k}} e'_{ij} A_i + e'_{kj} A_k. \quad (3.34)$$

Вектор A_r может быть разложен по векторам нового базиса. Формула (3.20) определяет соответствующие коэффициенты разложения. Подставим значение A_r в (3.34) и сгруппируем составляющие при одинаковых векторах

$$\sum_{\substack{i \in B_1 \\ i \neq r}} \left(e_{ij} - \frac{z_{ik}}{z_{rk}} e_{rj} \right) A_i + \frac{e_{rj}}{z_{rk}} A_k = \sum_{\substack{i \in B_2 \\ i \neq k}} e'_{ij} A_i + e'_{kj} A_k.$$

Множества индексов $(i \in B_1, i \neq r)$ и $(i \in B_2, i \neq k)$ совпадают. Приравнявая коэффициенты при одних и тех же векторах в правой и левой частях последнего равенства, получим требуемые рекуррентные соотношения для e_{ij}

$$e'_{ij} = e_{ij} - \frac{e_{rj}}{z_{rk}} z_{ik}, \quad i \neq k, i \in B; \quad (3.35)$$

$$e'_{kj} = \frac{e_{rj}}{z_{rk}}, \quad j = 1, 2, \dots, m. \quad (3.36)$$

Аналогичную рекуррентную формулу можно получить для компонент $\mu_j (j \in B)$. Имеем

$$\mu_j' = \sum_{i \in B_2} c_i e'_{ij} = \sum_{\substack{i \in B_2 \\ i \neq k}} c_i e'_{ij} + c_k e'_{kj}.$$

Учитывая вычисленные сейчас соотношения, найдем

$$\mu_j' = \sum_{\substack{i \in B_1 \\ i \neq r}} c_i \left(e_{ij} - \frac{e_{rj}}{z_{rk}} z_{ik} \right) + c_k \frac{e_{rj}}{z_{rk}}.$$

Символ $i \neq r$ под знаком суммы можно опустить, поскольку r -е слагаемое равно нулю. Получаем

$$\mu_j' = \sum_{i \in B_1} c_i e_{ij} - \frac{e_{rj}}{z_{rk}} \sum_{i \in B_1} c_i z_{ik} + c_k \frac{e_{rj}}{z_{rk}}.$$

Формулы (3.32) и (3.9) позволяют переписать последнее соотношение в виде

$$\mu_j' = \mu_j - \frac{e_{rj}}{z_{rk}} (z_k - c_k)$$

или в принятых ранее обозначениях

$$\mu_j' = \mu_j - \frac{e_{rj}}{z_{rk}} \Delta_k, \quad j = 1, 2, \dots, m. \quad (3.37)$$

Обозначая Δ_k через z_{0k} , а μ_j через e_{0j} , можно придать соотношениям (3.35) — (3.37) единую форму записи. Рекуррентные формулы (3.35), (3.36) и (3.37) являются основой II алгоритма метода последовательного улучшения плана.

§ 6. II АЛГОРИТМ

6.1. Полученные выше формулы сводят решение задачи линейного программирования к составлению последовательности основных таблиц и одной вспомогательной таблицы. Первая основная таблица содержит составляющие разложения вектора ограничений B и единичных векторов e_i по векторам исходного базиса B_1 . При описании первого алгоритма мы приняли обозначение $x_i = z_{i0}$. Такое решение было продиктовано аналогией между рекуррентными формулами, связывающими

ми компоненты разложения вектора ограничений и векторов условий по векторам двух последовательных базисов. Во втором алгоритме такие же причины определяют целесообразность обозначения $x_i = e_{i0}$ и соответственно $B = e_0$. Вычисления по II алгоритму начинаются с составления табл. Ia.

Пусть для определенности начальный базис содержит векторы условий $A_1, A_2, \dots, A_m$. Эти векторы перечислены в табл. Ia в столбце B_1 («базис»). Слева от них указаны соответствующие значения коэффициентов линейной формы c_i .

Т а б л и ц а Ia

c_i	B_1	e_l							A_k
		e_0	e_1	e_2	$\dots$	e_l	$\dots$	e_m	
c_1	A_1	e_{10}	e_{11}	e_{12}	$\dots$	e_{1l}	$\dots$	e_{1m}	z_{1k}
c_2	A_2	e_{20}	e_{21}	e_{22}	$\dots$	e_{2l}	$\dots$	e_{2m}	z_{2k}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
c_r	A_r	e_{r0}	e_{r1}	e_{r2}	$\dots$	e_{rl}	$\dots$	e_{rm}	z_{rk}
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\dots$	$\vdots$	$\dots$	$\vdots$	$\vdots$
c_m	A_m	e_{m0}	e_{m1}	e_{m2}	$\dots$	e_{ml}	$\dots$	e_{mm}	z_{mk}
	μ	μ_0	μ_1	μ_2	$\dots$	μ_l	$\dots$	μ_m	Δ_k

В столбце e_0 содержатся компоненты исходного опорного плана ($x_i = e_{i0}$). Здесь, как и в первом алгоритме, исходный опорный план предполагается заданным. Вообще говоря, вычисление x_i связано с решением системы линейных уравнений. Однако в тех случаях, когда начальный базис составляется из единичных векторов (как это бывает, например, при использовании М-метода) компоненты x_i оказываются равными соответствующим составляющим вектора ограничений.

Столбцы $e_1, e_2, \dots, e_m$ содержат компоненты e_{ij} разложения единичных векторов e_j по векторам базиса.

Мы уже отмечали, что вычисление e_{ij} сводится к обращению матрицы векторов базиса

$$\|a_{ij}\| = (A_1, A_2, \dots, A_m) = \left\| \begin{array}{cccc} a_{11} & \dots & a_{1m} \\ \cdot & \cdot & \cdot \\ a_{m1} & \dots & a_{mm} \end{array} \right\|.$$

Обращение матрицы является, пожалуй, наиболее громоздкой операцией при составлении табл. Ia. Однако если начальный базис составлен из единичных векторов, то матрица e_{ij} также оказывается единичной. Столбец e_i содержит единицу на i -м месте. Все остальные составляющие e_i — нули.

Последняя строка табл. Ia заполняется компонентами μ_s . Эти величины вычисляются как суммы произведений элементов левого столбца c_i на соответствующие составляющие вектора e_s :

$$\mu_s = \sum_{i=1}^m c_i e_{is}.$$

Как легко видеть, значение μ_0 совпадает со значением линейной формы для выбранного опорного плана.

Последний столбец табл. Ia представляет собой разложение вектора A_k , который следует ввести в базис B_2 , по элементам исходного базиса B_1 . Коэффициенты разложения A_k по векторам B_1 равны z_{ik} . Для того чтобы выяснить, какой вектор следует ввести в базис, и вычислить параметры z_{ik} , необходимо составить вспомогательную таблицу — табл. II.

В столбцах $A_1, A_2, \dots, A_n$ табл. II перечислены составляющие a_{ij} всех векторов условий. Все эти составляющие заданы условиями задачи; $(m+1)$ -я строка табл. II содержит величины z_j для всех векторов условий. Значения z_j вычисляются по формуле

$$z_j = \sum_{i=1}^m \mu_i a_{ij} \quad (j = 1, 2, \dots, n).$$

Для удобства вычислений целесообразно переписать последнюю строку табл. II (кроме элемента μ_0) в качестве столбца табл. II вслед за последним вектором условий.

Суммы произведений элементов столбца μ на соответствующие компоненты вектора A_j дают необходимые для использования критерия оптимальности значения z_j .

Т а б л и ц а II

N	c_j												
	c_1	c_2	c_3		c_j		c_k		c_n				
	A_j									μ	μ'	μ''	...
	A_1	A_2	A_3		A_j		A_k		A_n				
1	a_{11}	a_{12}	a_{13}		a_{1j}		a_{1k}		a_{1n}	μ_1	μ_1'	μ_1''	...
..	..	...	..		..		..		..	..	..	..	...
i	a_{i1}	a_{i2}	a_{i3}		a_{ij}		a_{ik}		a_{in}	μ_i	μ_i'	μ_i''	...
..	..	..	..		..		..		..	..	..	..	..
m	a_{m1}	a_{m2}	a_{m3}		a_{mj}		a_{mk}		a_{mn}	μ_m	μ_m'	μ_m''	...
z_j	z_1	z_2	z_3		z_j		z_k		z_n				
Δ	Δ_1	Δ_2	Δ_3		Δ_j		Δ_k		Δ_n				
Δ'	Δ_1'	Δ_2'	Δ_3'		Δ_j'		Δ_k'		Δ_n'				
Δ''	Δ_1''	Δ_2''	Δ_3''		Δ_j''		Δ_k''		Δ_n''				
.	.	.	.		.		.		.				
.	.	.	.		.		.		.				
.	.	.	.		.		.		.				

Вычитая из значений z_j соответствующие коэффициенты линейной формы c_j , получим следующую строку табл. II — величины оценок Δ_j ($j=1, 2, \dots, n$). Если все элементы строки неотрицательны, то выбранный опорный план является решением задачи линейного программирования. Появление среди разностей Δ_j отрицательных чисел означает, что базис должен быть изменен и вычисления должны быть продолжены. Вектор A_k , соответствующий наименьшему значению Δ_j , должен быть введен в базис.

Компоненты разложения вектора A_k по векторам исходного базиса z_{ik} вычисляются как суммы произведе-

ний элементов i -й строки табл. Ia на составляющие k -го столбца табл. II.

$$z_{ik} = \sum_{s=1}^m e_{is} a_{sk}.$$

Величины z_{ik} вписываются в последний столбец табл. Ia. Для последующих преобразований основной табл. Ia удобно поместить в ее нижний правый угол вычисленную ранее величину Δ_k . Дальнейшие вычисления состоят из перехода от табл. Ia к табл. Ib и последующим и в расширении вспомогательной табл. II. Переход к табл. Ib производится по рекуррентным формулам (3.35), (3.36) и (3.37). Табл. II расширяется вправо за счет добавления столбцов $\mu', \mu'', \dots$ и вниз за счет строк $\Delta', \Delta'', \dots$ до тех пор, пока не будет получена строка $\Delta^{(+)}$, не содержащая отрицательных элементов.

Рассмотрим, как составляется табл. Ib.

В новый базис (B_2) вводится вектор A_k . Из базиса B_1 , как и в I алгоритме, удаляется вектор A_r , для которого

$$\frac{x_r}{z_{rk}} = \min_{\substack{i \in B_1 \\ z_{ik} > 0}} \frac{x_i}{z_{ik}}$$

или, что то же самое,

$$\frac{e_{r0}}{z_{rk}} = \min_{\substack{i \in B_1 \\ z_{ik} > 0}} \frac{e_{i0}}{z_{ik}},$$

e_{i0} и z_{ik} — элементы из столбцов e_0 и A_k табл. Ia. (Неравенство $z_{ik} < 0$ для всех $i \in B_1$ означает, что линейная форма не ограничена сверху на множестве планов задачи.)

Табл. Ib содержит коэффициенты разложения вектора ограничений $B = e_0$ и единичных векторов e_i по векторам нового базиса B_2 . Последняя строка табл. Ib — это составляющие вектора μ' . Последний столбец (в нем появляется необходимость, если опорный план, соответствующий базису B_2 , еще не оптимальный план) заполняется коэффициентами разложения вектора $A_{k'}$, по векторам B_2 и величиной $\Delta'_{k'}$. $A_{k'}$ — вектор, который должен заменить один из векторов базиса B_2 и увеличить при этом значение линейной формы. Все элементы табл. Ib

(за исключением элементов последнего столбца) заполняются на основе табл. Ia в соответствии с рекуррентными формулами (3.35), (3.36) и (3.37).

Строка A_k табл. Ib (соответствующая вновь введенному в базис вектору) образуется из строки A_r табл. Ia делением всех ее элементов на величину z_{rk} — последний элемент этой строки. Для вычисления любой другой строки табл. Ib (в том числе и строки μ) следует из соответствующей ей строки табл. Ia вычесть строку A_k табл. Ib, предварительно умноженную на последний элемент (расположенный в столбце A_k) преобразуемой строки.

Таблица Ib

c_i	B_2	e_l							A_k'
		e_0	e_1	e_2		e_l		e_m	
c_1	A_1	e'_{10}	e'_{11}	e'_{12}		e'_{1l}		e'_{1m}	$z_{1k'}$
..	...	..	...	..		..		..	...
c_{r-1}	A_{r-1}	$e'_{r-1,0}$	$e'_{r-1,1}$	$e'_{r-1,2}$		$e'_{r-1,l}$		$e'_{r-1,m}$	$z_{r-1,k'}$
c_k	A_k	e'_{k0}	e'_{k1}	e'_{k2}		e'_{kl}		e'_{km}	$z_{kk'}$
c_{r+1}	A_{r+1}	$e'_{r+1,0}$	$e'_{r+1,1}$	$e'_{r+1,2}$		$e'_{r+1,l}$		$e'_{r+1,m}$	$z_{r+1,k'}$
..	..	..	..	...		..		..	...
c_m	A_m	e'_{m0}	e'_{m1}	e'_{m2}		e'_{ml}		e'_{mm}	$z_{mk'}$
	μ'	μ'_0	μ'_1	μ'_2		μ'_l		μ'_m	$\Delta_{k'}$

Для контроля вычислений целесообразно получать μ'_s не только по рекуррентной формуле (3.37), но и непосредственно, пользуясь равенством

$$\mu'_s = \sum_{l \in B_2} c_l e'_{ls}.$$

После заполнения табл. Iб (кроме ее последнего столбца $A_{k'}$) следует снова обратиться к вспомогательной табл. II. Табл. II, как уже указывалось, служит для проверки на оптимальность очередного опорного плана.

Перенесем последнюю строку табл. Iб (за исключением элемента μ'_0) в очередной столбец табл. II — вслед за столбцом μ — и вычислим для всех векторов условий значения

$$z'_j = \sum_{i=1}^m \mu'_i a_{ij}.$$

Разности $\Delta'_j = z'_j - c_j$ образуют новую строку табл. II. По этой строке второй опорный план проверяется на оптимальность. Если он не оптимален, то вычисления продолжаются и от II плана переходят к III точно так же, как от I плана перешли ко II.

Как видим, известные трудности представляет только составление табл. Ia и то лишь в том случае, если исходный базис B_1 составлен не из единичных векторов. Переход от одной из основных таблиц к следующей совершается, главным образом, по рекуррентным формулам. Несколько более громоздкие вычисления требуются для определения очередных значений

$$z_{ik}^{(s)} \text{ и } \Delta_j^{(s)}.$$

Каждый следующий шаг во II алгоритме состоит в вычислении новой таблицы типа Iб и добавлении строки оценок $\Delta_j^{(s)}$ во вспомогательную табл. II.

В невырожденной задаче линейного программирования значение линейной формы (левый нижний элемент основных таблиц $\mu_0^{(s)}$) возрастает на каждом шагу. Заметим, что $\Delta_0^{(s)} = \mu_0^{(s)} = L(X^{(s)})$.

Вычисления прекращаются, когда все Δ_j оказываются неотрицательными и, следовательно, достигается конечный максимум $L(x)$, или когда устанавливается неограниченность линейной формы на множестве планов рассматриваемой задачи.

6.2. Приведем сравнительную оценку двух изложенных алгоритмов. Начнем с того, что решим, пользуясь вторым алгоритмом, те же примеры, которые были рассмотрены при описании первого алгоритма вычислений.

Пример 1. Перепишем условия задачи, переведенной уже в каноническую форму. Требуется обратить в максимум линейную форму

$$L = -2x_1 - 3x_2 + 2x_4 + 3x_5$$

при условиях:

$$\begin{aligned} 2x_1 - x_2 + x_3 + x_6 &= 12, \\ x_1 + x_4 + x_7 &= 5, \\ 2x_1 + 2x_2 + x_3 - 2x_5 + x_8 &= 20, \\ x_1 - x_2 - 2x_3 + 2x_4 - 2x_5 + x_9 &= 10, \\ -2x_1 + 2x_2 - 2x_3 - 2x_4 + x_5 + x_{10} &= 24, \\ x_i &\geq 0, \quad i = 1, 2, \dots, 10. \end{aligned}$$

В качестве начального опорного плана выбираем, как и прежде, вектор $X = (0, 0, 0, 0, 0, 1, 2, 5, 20, 10, 24)$. Начальный базис образуется векторами $A_6, \dots, A_{10}$.

Составление табл. Ia не представляет труда. Исходный базис определяет единичную матрицу $\|a_{ij}\|$. Поэтому и обратная матрица $\|e_{ij}\|$ будет единичной, а составляющие разложения вектора ограничений по векторам базиса равны соответствующим ограничениям b_i .

Таблица Ia

c_i	B_i	e_i						A_b
		e_0	e_1	e_2	e_3	e_4	e_5	
	A_6	12	1					
	A_7	5		1				
	A_8	20			1			-2
	A_9	10				1		-2
	A_{10}	24					1	1
μ								-3

Векторы базиса отвечают дополнительным переменным и входят в линейную форму с нулевыми коэффициентами. Поэтому соответствующие значения

$$\mu_s = \sum_{i \in B_i} c_i e_{is} = 0.$$

Составим вспомогательную табл. II и внесем в неё составляющие a_{ij} всех векторов условий A_j ($j = 1, 2, \dots, 10$). После векторов условий выписываются составляющие вектора μ . В нашем случае это нули. Поэтому все $z_j = \sum_{i=1}^5 \mu_i a_{ij} = 0$, и в строке Δ_j записываются значения $\Delta_j = z_j - c_j = -c_j$. Наименьшее значение Δ_j равно -3 . Эта величина соответствует вектору A_5 . Его и следует ввести в базис B_2 . Компоненты разложения A_5 по векторам базиса B_1 выписываются в последнюю колонку табл. Ia. В рассматриваемом примере составляющие A_5 по векторам базиса B_1 равны соответствующим компонентам A_5 , поскольку базис составлен из единичных векторов: $z_{i5} = a_{i5}$.

Т а б л и ц а II

N						-2	-3		2	3				
	A_6	A_7	A_8	A_9	A_{10}	A_1	A_2	A_3	A_4	A_5	μ	μ'	μ''	μ'''
1	1					2	-1	1						6
2		1				1			1				8	8
3			1			2	2	1		-2				
4				1		1	-1	-2	2	-2				
5					1	-2	2	-2	-2	1		3	3	3
Δ						2	3		-2	-3				
Δ'					3	-4	9	-6	-8					
Δ''		8			3	4	9	-6						
Δ'''	6	8			3	16	3							

Среди величин z_{i5} только одно положительное число. Это коэффициент при A_{10} в разложении A_5 по базису B_1 . Следовательно, вектор A_5 должен быть введен в базис вместо A_{10} . Теперь можно перейти к составлению табл. Ib. Заполним строку, соответствующую новому

вектору базиса. Воспользуемся рекуррентной формулой (3.36)

$$e'_{kj} = \frac{e_{rj}}{z_{rk}}.$$

В нашем случае $z_{rk} = z_{10,5} = 1$. Следовательно, $e'_{kj} = e_{rj}$.

Т а б л и ц а 16

c_i	B_j	e_l						A_k
		e_0	e_1	e_2	e_3	e_4	e_5	
	A_6	12	1					
←	A_7	5		1				1
•	A_8	68			1		2	−4
	A_9	58				1	2	−2
→	3 A_5	24					1	−2
	μ'	72					3	−8

Чтобы получить остальные составляющие векторов $e_0, e_1, \dots, e_5$, воспользуемся следующей записью формулы (3.35):

$$e'_{ij} = e_{ij} - e'_{kj} z_{ik}.$$

Отсюда видно, что для получения компоненты e_j в новом базисе следует из соответствующей компоненты в старом базисе вычесть произведение элемента той же строки (i), но последнего столбца (A_k) табл. 1а и элемента того же столбца (j), но вновь введенной строки (A_k) табл. 1б. Например,

$$e'_{8,0} = e_{8,0} - e'_{5,0} \cdot z_{8,5} = 20 - 24 \cdot (-2) = 68,$$

$$e'_{9,4} = e_{9,4} - e'_{5,4} \cdot z_{9,5} = 1 - 0 \cdot (-2) = 1,$$

$$e'_{8,5} = e_{8,5} - e'_{5,5} \cdot z_{8,5} = 0 - 1 \cdot (-2) = 2.$$

Так могут быть получены все значения e'_{ij} .

Элементы последней строки (μ) вычисляются либо по рекуррентной формуле (3.37), либо непосредственно из равенства (3.32).

В рассматриваемом примере только один вектор базиса B_2 отвечает ненулевому коэффициенту линейной формы. Это вектор A_5 , для которого $C_5 = 3$. Отсюда $\mu'_0 = 3 \cdot e_{5,0} = 72$, $\mu'_5 = 3 \cdot 1 = 3$. Все остальные μ'_s равны нулю. То же получаем по рекуррентной формуле. Например,

$$\mu'_0 = \mu_0 - e'_{5,0} \cdot \Delta_5 = 0 - 24 \cdot (-3) = 72,$$

$$\mu'_3 = \mu_3 - e'_{5,3} \cdot \Delta_5 = 0 - 0 \cdot (-3) = 0.$$

Полученные таким образом значения μ'_s вносятся в очередной столбец вспомогательной табл. II. В нашем примере в колонку μ' вносится только одно число $\mu'_5 = 3$.

Теперь следует получить величины z'_j ($z'_j = \sum_{i=1}^m \mu'_i a_{ij}$).

В нашем случае $z'_j = \mu'_5 a_{5j} = 3a_{5j}$.

Вписываем очередную строку в табл. II:

$$\Delta'_j = z'_j - c_j.$$

Как видим, имеются отрицательные значения Δ'_j . Следовательно, базис B_2 не соответствует оптимальному опорному плану. В строке Δ'_j наименьший элемент это $\Delta'_4 = -8$. Поэтому в очередной базис следует ввести A_4 . Значения z'_{i4} — составляющие разложения A_4 по векторам базиса B_2 — вносятся в последнюю колонку табл. Ib. Они вычисляются по формуле

$$z'_{i4} = \sum_{s=1}^m a_{s4} e'_{is}.$$

Здесь a_{s4} — элементы столбца A_4 табл. II, а e'_{is} — компоненты i -й строки табл. Ib. Далее следует определить, какой вектор необходимо вывести из базиса. После этого переходят от табл. Ib к табл. Iv и от табл. Iv к следующей (Iг) точно так же, как переходили от табл. Ia к табл. Ib. При этом вспомогательная таблица окаймляется справа столбцами $\mu^{(s)}$ и снизу строками $\Delta^{(s)}$

Таблица Ів

c_l	B_3	e_l						A_3
		e_0	e_1	e_2	e_3	e_4	e_5	
←	A_6	12	1					1
→	2	A_4	5		1			
	A_8	88		4	1		2	—3
	A_9	68		2		1	2	—6
3	A_5	34		2			1	—2
μ''		112		8			3	—6

Таблица Іг

c_l	B_4	e_l						
		e_0	e_1	e_2	e_3	e_4	e_5	
→	A_3	12	1					
2	A_4	5		1				
	A_8	124	3	4	1		2	
	A_9	140	6	2		1	2	
3	A_5	58	2	2			1	
μ'''		184	6	8			3	

Строка Δ''' во вспомогательной табл. ІІ состоит из неотрицательных чисел. Следовательно, опорный план, к которому мы пришли в табл. Іг, является решением задачи.

Пример 2. В основной системе табл. ІV и вспомогательной табл. ІІІ записаны все этапы решения примера 2, рассмотренного при описании І алгоритма. Здесь сохранены обозначения и нумерация, принятые в § 4.

Таблица III

c_i	A_{16}	A_{17}	A_{18}	A_{19}	A_{20}	A_1	A_2	A_3	A_4	A_5	A_6	A_7	A_8	A_9	A_{10}	A_{11}	A_{12}	A_{13}	A_{14}	A_{15}	μ'	μ''	μ'''
N																							
1	1					2	-1	1			1	1		1	-1	-2	-1	2		1			8
2		1				1			1		-1	1	2		1	1	1	1	1			8	8
3			1			2	2	1		-2	1	2	-1	1	-2		2	-2		2			
4				1		1	-1	-2	2	-2	1		-1		2	-2	-1	2					
5					1	-2	2	-2	-2	1	1	1	1		1	2	2		1	-2	3	3	3
Δ						2	3		-2	-3		1	1	-2	1	3	-2	1	-1	-2			
Δ'						-4	9	-6	-8		3	4	4	-2	4	9	4	1	2	-8			
Δ''						4	9	-6			-5	12	20	-2	12	17	12	1	10	-8			
Δ'''						20	1	2			3	20	20	6	4	11	4	17	10				

Таблица IV

c_j	B	e_l						A_k
		e_0	e_1	e_2	e_3	e_4	e_5	
	A_{16}	12	1					
	A_{17}	5		1				
	A_{18}	20			1			-2
	A_{19}	10				1		-2
←	A_{20}	24					1	1
	μ							-3
	A_{16}	12	1					
←	A_{17}	5		1				1
	A_{18}	68			1		2	-4
	A_{19}	58				1	2	-2
→	3 A_5	24					1	-2
	μ'	72					3	-8
←	A_{16}	12	1					1
→	2 A_4	5		1				
	A_{18}	88		4	1		2	-2
	A_{19}	68		2		1	2	-4
	3 A_5	34		2			1	-2
	μ''	112		8			3	-8
→	2 A_{15}	12	1					
	2 A_4	5		1				
	A_{18}	112	2	4	1		2	
	A_{19}	116	4	2		1	2	
	3 A_5	58	2	2			1	
	μ'''	208	8	8			3	

Оценивая сравнительные достоинства и недостатки двух вычислительных алгоритмов, реализующих метод последовательного улучшения плана, необходимо отметить следующее.

Наиболее трудоемкая работа в каждом алгоритме — это составление исходных основных таблиц. Обычно эта работа бывает связана с решением систем линейных уравнений или с обращением матриц. Вычислительные трудности при составлении начальных таблиц исключаются, если исходный базис состоит из единичных векторов.

В тех случаях, когда число условий существенно меньше числа переменных, каждый шаг II алгоритма требует значительно меньше операций, чем соответствующий этап в I алгоритме. Этот вывод справедлив, несмотря на то, что во II алгоритме появляется дополнительная потребность во вспомогательной таблице.

Общее число шагов в обоих алгоритмах, естественно, одно и то же.

II алгоритм следует рекомендовать для решения задач, у которых матрица условий содержит много нулей. В этом случае расчеты существенно сокращаются.

Отдельные операции во II алгоритме (например, вычисление оценок $\Delta_j^{(s)}$ и компонент z_{ik} вектора, вводимого в базис) несколько сложнее элементарных операций I алгоритма. Тем не менее во всех случаях, когда исходный базис является искусственным базисом (т. е. состоит из единичных векторов) и число переменных в несколько раз превышает число условий, следует рекомендовать вычисление оптимального плана по II алгоритму. Пример 2, решенный здесь в соответствии с тем и другим алгоритмом, подтверждает этот вывод. Второй алгоритм дает решение двух взаимно сопряженных задач. В ряде случаев это оказывается полезным.

В тех задачах, в которых число переменных и число условий — величины одного порядка, необходимость во вспомогательной таблице при II алгоритме заставляет отдавать предпочтение I алгоритму.

В заключение следует отметить, что метод последовательного улучшения плана является наиболее испытанным в практике вычислений. Опыт показывает, что количество итераций обычно колеблется в пределах от

m до $2m$. Однако до сих пор еще не получены достаточно хорошие теоретические оценки трудоемкости метода.

6.3. Приведем теперь пример использования M -метода для решения задачи линейного программирования, в которой исходный план неочевиден. Вычисления будут проведены в соответствии со II алгоритмом метода последовательного улучшения плана.

Пусть требуется обратить в максимум линейную форму

$$L(X) = x_1 + 2x_2 + 3x_3 + x_4 + 2x_5 + 3x_6 + x_7$$

при условиях:

$$\begin{aligned} x_1 + x_2 + 2x_3 + 3x_4 + 2x_5 + 3x_6 + x_7 &= 7, \\ 2x_1 + 3x_2 + x_3 + x_4 + 3x_5 + 2x_6 + 2x_7 &= 8, \\ x_1 + 2x_2 + 3x_3 + 2x_4 + 0,5x_5 + x_6 + 2x_7 &= 6, \\ 2x_1 + x_2 + 3x_3 + x_4 + 2x_5 + 3x_6 + 2x_7 &= 7, \\ x_i &\geq 0, \quad (i=1, 2, \dots, 7). \end{aligned}$$

Начальный опорный план заранее не задан. Введем дополнительные неотрицательные переменные x_8, x_9, x_{10} и x_{11} и рассмотрим соответствующую M -задачу.

Требуется обратить в максимум линейную форму

$$\tilde{L} = L(X) - M(x_8 + x_9 + x_{10} + x_{11})$$

при следующих ограничениях:

$$\begin{aligned} L_1(X) + x_8 &= 7, \\ L_2(X) + x_9 &= 8, \\ L_3(X) + x_{10} &= 6, \\ L_4(X) + x_{11} &= 7, \\ x_i &\geq 0 \quad (i=1, 2, \dots, 11). \end{aligned}$$

Здесь $L(X)$ — линейная форма, а $L_i(X)$ — левые части условий исходной задачи.

Обозначим искусственные векторы условий через e_1, e_2, e_3 и e_4 соответственно. Они, очевидно, составляют начальный базис M -задачи.

Построим в соответствии с правилами II алгоритма табл. Ia. Начальный базис определяет единичную матрицу. Поэтому и матрица $\|e_{ij}\|$ будет единичной, а составляющие разложения вектора ограничений $B = e_0$ по

векторам базиса равны соответствующим ограничениям b_i .

Т а б л и ц а Iа

c_i	B_1	e_i					A_s	$\frac{e_{i0}}{e_{i3}}$
		e_0	e_1	e_2	e_3	e_4		
$-M$	e_1	7	1				2	$7/2$
$-M$	e_2	8		1			1	8
$-M$	e_3	6			1		3	2
$-M$	e_4	7				1	3	$7/3$
μ		$-28M$	$-M$	$-M$	$-M$	$-M$	$-9M-3$	

Нижняя строка таблицы заполняется значениями μ_s

$$\mu_s = \sum_{i \in B_1} c_i e_{is}.$$

Составим вспомогательную табл. II.

Таблица содержит составляющие a_{ij} векторов условий $A_1, A_2, \dots, A_7$ и компоненты вектора μ . В строке Δ записываются значения Δ_s

$$\Delta_s = \sum_{i=1}^m \mu_i a_{is} - c_s.$$

Значения Δ_s , как и величины μ , являются линейными функциями параметра M .

Наименьшая из величин Δ_s (наибольшая по абсолютной величине) определяет вектор, который следует ввести в базис. Поскольку $M > 0$ по условию сколь угодно большое число, сравнение величин Δ_s производится по следующему правилу:

$$\Delta_{s_1} = \alpha_1 M + \beta_1 > \Delta_{s_2} = \alpha_2 M + \beta_2, \text{ если } \alpha_1 > \alpha_2.$$

При $\alpha_1 = \alpha_2$ $\Delta_{s_1} > \Delta_{s_2}$, если $\beta_1 > \beta_2$.

В рассматриваемом примере наименьшее значение Δ_s равно

$$\Delta_3 = \Delta_6 = -9M - 3.$$

Таблица II

A_j	A_1	A_2	A_3	A_4	A_5	A_6	A_7	μ	μ'	μ''	μ'''	μ^{IV}
c_i	c_j											
	1	2	3	1	2	3	1					
$-M$	1	1	2	3	2	3	1	$-M$	$-M$	$-M$	$-M$	$7-\frac{7}{48}$
$-M$	2	3	1	1	3	2	2	$-M$	$-M$	$-M$	$\frac{5M+6}{19}$	$\frac{17}{48}$
$-M$	1	2	3	2	0,5	1	2	$-M$	$2M+1$	$-M$	$\frac{-7M+3}{19}$	$\frac{5}{48}$
$-M$	2	1	3	1	2	3	2	$-M$	$-M$	$2M+1$	$\frac{18M+14}{19}$	$\frac{7}{8}$
Δ	$-6M-1$	$-7M-2$	$-9M-3$	$-7M-1$	$-7,5M-2$	$-9M-3$	$-7M-1$					
Δ'	$-3M$	$-M$		$-M+1$	$-6M-1,5$	$-6M-2$	$-M+1$					
Δ''	1	$-4M-1$		$-4M$	$-1,5M$		$-M+1$					
Δ'''	$\frac{20M+24}{19}$			$\frac{-48M+7}{19}$	$\frac{M+1}{2}$		$\frac{13M+27}{19}$					
Δ^{IV}	$\frac{17}{2}$				$\frac{55}{96}$		$\frac{73}{48}$					

Поэтому в качестве вектора, который следует ввести в базис, можно выбрать A_3 или A_6 . Введем в базис вектор A_3 . Компоненты A_3 выписываются в предпоследний столбец табл. Ia. Последним элементом этого столбца записывается величина Δ_3 . Все составляющие A_3 положительны. В последний столбец табл. Ia вносятся отношения элементов столбца e_0 к соответствующим положительным компонентам вектора A_3 . Сравнивая величины этих отношений, находим индекс вектора, который следует вывести из базиса B_1 : Наименьший элемент последнего столбца равен 2. Он соответствует вектору e_3 . Следовательно, A_3 должен быть введен в базис вместо искусственного вектора e_3 .

Теперь можно перейти к составлению табл. Ib.

Т а б л и ц а 1б

c_i	B_i	e_i					A_6	$\frac{e_{i0}}{e_{i6}}$
		e_0	e_1	e_2	e_3	e_4		
$-M$	e_1	3	1		$-2/3$		$7/3$	$9/7$
$-M$	e_2	6		1	$-1/3$		$5/3$	$18/5$
$\rightarrow 3$	A_3	2			$1/3$		$1/3$	6
$\leftarrow -M$	e_4	1			-1	1	2	$1/2$
μ'		$-10M+6$	$-M$	$-M$	$2M+1$	$-M$	$-6M-2$	

Элементы таблицы вычисляются по рекуррентным формулам (3.35) — (3.37) точно так же, как это делалось в предыдущем примере. Составляющие вектора μ' переписываются, кроме того, в очередной столбец вспомогательной табл. II. Компоненты очередной строки Δ' табл. II вычисляются по формуле

$$\Delta_j' = z_j' - c_j = \sum_{i=1}^m \mu_i' a_{ij} - c_j.$$

Наименьшим элементом последней строки является $\Delta_6' = -6M - 2$. Следовательно, в новый базис следует

ввести вектор A_6 . Коэффициенты разложения A_6 по новому базису вычисляются по формулам

$$z'_{i6} = \sum_{s=1}^4 a_{s6} e'_{is}.$$

Здесь a_{s6} — составляющие A_6 из табл. II,
 e'_{is} — соответствующие элементы табл. Iб.

Вычисленные таким образом величины z'_{i6} записываются в предпоследний столбец табл. Iб. Последний элемент этого столбца Δ'_6 удобно для контроля вычислить по формуле

$$\Delta'_6 = \sum_{i=1}^4 c_i z'_{i6} - c_6 = -6M - 2.$$

В крайнем справа столбце содержатся отношения $\frac{e_{i0}}{e'_{i6}}$, необходимые для выявления вектора, который должен быть исключен из базиса. Таким вектором является e_4 .

Теперь следует от табл. Iб перейти к табл. Iв и от табл. Iв к последующим по тем же правилам, по которым табл. Ia преобразовывалась в табл. Iб. При этом вспомогательная таблица окаймляется справа столбцами μ^s и снизу строками Δ^s .

Строка $\Delta^{(IV)}$ в табл. II состоит из неотрицательных чисел. Следовательно, опорный план, к которому приходим в табл. Id, является решением задачи.

Оптимальный план рассматриваемой задачи равен

$$X = \left\{ 0, \frac{25}{16}, \frac{3}{8}, \frac{3}{16}, 0, \frac{11}{8}, 0 \right\}.$$

Максимальное значение линейной формы равно 137/16.

В рассматриваемом примере оказалось, что после исключения всех искусственных векторов из базиса план сразу оказался оптимальным. В общем случае это может быть не так. После удаления из базисов векторов e_i обычно получается опорный план, который в процессе последующего улучшения должен быть преобразован в оптимальный план.

Заметим, что в ходе решения задачи мы придерживались следующего правила. Искусственный вектор,

Таблицы Ів, Іг, Ід

c_i	$\frac{e_i}{B_s}$	e_0	e_1	e_2	e_3	e_4	A_k	$\frac{e_{i0}}{e_{ik}}$
$-M$	e_1	$11\frac{1}{6}$	1		$1\frac{1}{2}$	$7\frac{1}{6}$	$5\frac{5}{6}$	$11\frac{1}{5}$
$-M$	e_2	$31\frac{1}{6}$		1	$1\frac{1}{2}$	$5\frac{5}{6}$	$19\frac{1}{6}$	$31\frac{1}{19}$
3	A_3	$11\frac{1}{6}$			$1\frac{1}{2}$	$1\frac{1}{6}$	$5\frac{5}{6}$	$11\frac{1}{5}$
3	A_6	$1\frac{1}{2}$			$1\frac{1}{2}$	$1\frac{1}{2}$	$1\frac{1}{2}$	
		$-7M+7$	$-M$	$-M$	$-M$	$2M+1$	$-4M-1$	
$-M$	e_1	$9\frac{9}{19}$	1	$5\frac{5}{19}$	$7\frac{7}{19}$	$18\frac{18}{19}$	$48\frac{48}{19}$	$9\frac{9}{48}$
2	A_2	$31\frac{31}{19}$		$6\frac{6}{19}$	$3\frac{3}{19}$	$5\frac{5}{19}$	$7\frac{7}{19}$	$31\frac{31}{7}$

←

→

←

→

Продолжение табл. Ів, Іг, Ід

c_l	$\begin{matrix} e_l \\ B_S \end{matrix}$	e_0	e_1	e_2	e_3	e_4	A_k	$\frac{e_{l0}}{e_{lk}}$
3	A_3	$\frac{9}{19}$		$-\frac{5}{19}$	$\frac{7}{19}$	$\frac{1}{19}$	$\frac{10}{19}$	$\frac{9}{10}$
3	A_6	$\frac{25}{19}$		$\frac{3}{19}$	$-\frac{8}{19}$	$\frac{7}{19}$	$-\frac{6}{19}$	
		$\frac{-9M+164}{19}$	$-M$	$\frac{5M+6}{19}$	$\frac{-7M+3}{19}$	$\frac{18M+14}{19}$	$\frac{-48M+7}{19}$	
1	A_4	$\frac{3}{16}$	$\frac{19}{48}$	$-\frac{5}{48}$	$\frac{7}{48}$	$-\frac{3}{8}$		
2	A_2	$\frac{25}{16}$	$-\frac{7}{48}$	$\frac{17}{48}$	$\frac{5}{48}$	$-\frac{1}{8}$		
3	A_3	$\frac{3}{8}$	$-\frac{5}{24}$	$-\frac{5}{24}$	$\frac{7}{24}$	$\frac{1}{4}$		
3	A_6	$\frac{11}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{3}{8}$	$\frac{1}{4}$		
		$\frac{137}{16}$	$-\frac{7}{48}$	$\frac{17}{48}$	$\frac{5}{48}$	$\frac{7}{8}$		

→

выведенный из базиса, исключался из дальнейшего рассмотрения. Поэтому искусственные векторы не вводились во вспомогательную таблицу. Нетрудно видеть, что приведенное правило, сокращая объем вычислений, не меняет сущности M -метода.

Как и ранее, мы всегда приходим к решению задачи или убеждаемся в ее неразрешимости.

§ 7. ВЫРОЖДЕННОСТЬ

7.1. В ряде предыдущих рассуждений было использовано допущение о невырожденности задачи линейного программирования. Невырожденность означает, что каждый опорный план задачи содержит ровно m положительных компонент

В геометрической интерпретации задачи линейного программирования для простейшего случая, когда $n - m = 2$ (гл. 1, § 4), легко отличить вырожденную задачу линейного программирования от невырожденной. В вырожденной задаче в одной вершине многоугольника пересекается более двух прямых, описываемых уравнениями вида $x_k = 0$. Это значит, что одна или несколько сторон многоугольника условий стягиваются в точку. Аналогично при $n - m = 3$ в вырожденной задаче в одной вершине пересекается более трех плоскостей $x_k = 0$. Аналогия может быть продолжена и дальше.

Во второй геометрической интерпретации задачи линейного программирования (гл. 1, § 7) вырожденность (в случае $m = 2$) означает, что прямая $\mathcal{L}$ пересекает ребро, а не грань многогранного конуса. При $m > 2$ вырожденность означает, что некоторая общая точка прямой $\mathcal{L}$ и конуса задачи принадлежит многогранному конусу, порожденного не более чем $(m - 1)$ ребром.

В методе последовательного улучшения плана допущение о невырожденности существенно используется при переходе от одного базиса к другому. Понятие базиса было введено в § 8 гл. 1. Приведенное там определение относится к невырожденным задачам линейного программирования. В вырожденном случае понятие базиса должно быть уточнено. Под базисом опорного плана задачи линейного программирования, записанной в канонической форме, подразумевается система из m линейно независимых векторов условий, среди которых

содержатся все векторы при положительных компонентах плана.

В невырожденной задаче любой опорный план содержит ровно m положительных компонент. Векторы условий при этих компонентах однозначно определяют базис опорного плана. В вырожденной задаче некоторые опорные планы содержат $\nu < m$ положительных составляющих. В соответствии с приведенным определением базис такого плана образуется из ν векторов, отвечающих положительным компонентам, и $m - \nu$ векторов, связанных с нулевыми составляющими плана. Очевидно, что в этом случае базис определяется неоднозначно. Единственное требование, предъявляемое к дополнительным $m - \nu$ векторам, состоит в том, чтобы полученная система из m векторов была линейно независима.

В предположении о невырожденности существует только одно значение i ($i = r$), при котором достигается

$$\min_i \frac{x_i}{z_{ik}}, \quad z_{ik} > 0.$$

Это позволяет однозначно определить вектор A_r , который следует исключить из базиса и заменить вектором A_k . Вектор ограничений B снова разлагается по m векторам очередного базиса, и значение линейной формы при этом увеличивается. Приведенные рассуждения теряют силу, если окажется, что существует несколько значений i , при которых $\frac{x_i}{z_{ik}}$ (для $z_{ik} > 0$) обращается в минимум. При

$$\theta = \min_i \frac{x_i}{z_{ik}}, \quad z_{ik} > 0$$

в разложении (3.6) обратится в нуль несколько коэффициентов при векторах условий. Вектор ограничений в этом случае выразится менее чем через m векторов системы $A_1, A_2, \dots, A_n$. Задача линейного программирования вырождается.

Вычислительная процедура метода последовательного улучшения плана заключается в последовательном переходе от одного базиса к следующему. При этом с каждым шагом значение линейной формы увеличивается. В предположении о невырожденности произвольным является только выбор исходного базиса. Последующий переход от одного базиса к очередному совершается одно-

значно *). В вырожденном случае нет оснований для исключения из базиса того или иного из векторов с исчезающим (при $\theta = \min_{z_{ij} > 0} \frac{x_i}{z_{ij}}$) коэффициентом. Дело в том,

что переход от одного базиса к следующему в вырожденной задаче может не менять значения линейной формы. Это видно из формул (3.8) и (3.11). В вырожденном случае по крайней мере одна из m компонент x_i опорного плана, для которой $z_{ij} > 0$, может оказаться равной нулю. Формулы (3.8) показывают, что при этом единственным значением θ , которое приводит к новому базису, сохраняя $x'_k = x_k - \theta z_{ik}$ неотрицательными, является $\theta = 0$. А в таком случае, как видно из (3.11), $L(X') = L(X)$.

Сохранение величины линейной формы может быть геометрически истолковано следующим образом. В простейшем случае при $n - m = 2$ в вырожденной задаче две или несколько вершин многоугольника условий сливаются в одну. Ясно, что переход от одной из этих вершин к другой не может изменить значения линейной формы.

Та же геометрическая интерпретация может быть использована и при $n - m > 2$. В случае вырожденности две или больше вершин многогранника условий сливаются в одну и, следовательно, переход от одной из них к другой не меняет значения линейной формы.

В невырожденной задаче только увеличение линейной формы гарантировало от повторения базиса. В вырожденном случае возникает возможность цикла — возможность возвращения к базису, который уже встречался.

По-видимому, зацикливание является редким событием в практике линейного программирования. Построение примера вырожденной задачи, приводящей к зацикливанию, вызвало известные затруднения. Не занимаясь здесь разбором примера, имеющего главным образом теоретический интерес, отметим следующие доказанные при построении примера обстоятельства.

*) Предполагается, что в базис вводится вектор, соответствующий минимальному значению оценки Δ . Если таких векторов несколько, выбирается вектор с наименьшим номером.

Зацикливание невозможно, если вектор ограничений выражается не менее, чем через $(m - 1)$ векторов условий. Цикл может встретиться только в задачах, в которых возможно разложение вектора ограничений по $(m - 2)$ или менее векторам условий. При этом цикл не может включать менее шести шагов. Тем не менее принципиальная возможность цикла требует практических рекомендаций, обеспечивающих во всех случаях решение вырожденных задач линейного программирования.

7.2. Геометрическое истолкование явления вырожденности указывает пути устранения опасности зацикливания.

Вырожденность, как уже отмечалось, соответствует тому случаю, когда прямая $\mathcal{L}$ пересекает хотя бы один из многогранных конусов, порожденных не более чем $(m - 1)$ ребром. Следует ожидать, что, смещая несколько прямую $\mathcal{L}$ параллельно самой себе, можно исключить этот случай. Смещение прямой $\mathcal{L}$ связано с изменением вектора ограничений, однозначно определяющего ее положение. Итак, небольшое соответствующим образом подобранное изменение вектора ограничений может устранить опасность возникновения цикла. На этом основана идея рекомендации Чарнеса [25], гарантирующая во всех случаях решение задачи линейного программирования в конечное число шагов.

Пусть базис составлен из векторов условий $A_1, A_2, \dots, A_m$. В соответствии с формулами (3.4) и (3.7) разложение вектора ограничений и векторов условий по векторам базиса может быть представлено в виде

$$B = A_1 x_1 + \dots + A_m x_m, \quad (3.4)$$

$$A_j = A_1 z_{1j} + \dots + A_m z_{mj}. \quad (3.7)$$

Осуществим сдвиг вектора ограничений следующим образом. Умножим (3.7) на ϵ^j ($\epsilon > 0$), просуммируем полученные соотношения по j от 1 до n и сложим результат с соотношением (3.4). Получим

$$B + \sum_{j=1}^n \epsilon^j A_j = A_1 \left(x_1 + \sum_{j=1}^n \epsilon^j z_{1j} \right) + \dots + \\ + A_m \left(x_m + \sum_{j=1}^n \epsilon^j z_{mj} \right).$$

Введем обозначения

$$B(\varepsilon) = B + \sum_{j=1}^n \varepsilon^j A_j; \quad x_i(\varepsilon) = x_i + \sum_{j=1}^n \varepsilon^j z_{ij};$$

$$X(\varepsilon) = \{x_1(\varepsilon), \dots, x_m(\varepsilon)\}.$$

Тогда

$$B(\varepsilon) = A_1 x_1(\varepsilon) + \dots + A_m x_m(\varepsilon), \quad (3.38)$$

$$L[X(\varepsilon)] = c_1 x_1(\varepsilon) + \dots + c_m x_m(\varepsilon). \quad (3.39)$$

С каждой задачей линейного программирования (3.1) — (3.3) связано семейство так называемых ε -задач. В ε -задаче вычисляется экстремум линейной формы (3.39) при условиях

$$B(\varepsilon) = A_1 x_1(\varepsilon) + \dots + A_n x_n(\varepsilon)$$

и

$$x_j(\varepsilon) \geq 0 \quad (j = 1, 2, \dots, n.)$$

Правило исключения вектора из базиса, гарантирующее от заикливания в процессе решения, может быть выработано на основе следующих двух утверждений.

I. Существует такое малое $\varepsilon_1 > 0$, что для любого $\varepsilon < \varepsilon_1$, ε -задача является невырожденной.

II. Существует такое малое $\varepsilon_2 > 0$, что для любого $\varepsilon < \varepsilon_2$ всякому опорному плану ε -задачи соответствует опорный план исходной задачи. При этом оптимальному плану ε -задачи соответствует решение исходной задачи. (Опорные планы исходной задачи и ε -задачи называются соответствующими, если они имеют общий базис.)

Докажем приведенные утверждения. Пусть $A_{s_1}, A_{s_2}, \dots, A_{s_m}$ произвольная линейно независимая система векторов условий. Разложим $B(\varepsilon)$ по векторам этой системы

$$B(\varepsilon) = \sum_{i=1}^m x_{s_i}(\varepsilon) A_{s_i},$$

где
$$x_{s_i}(\varepsilon) = x_{s_i} + \sum_{j=1}^m \varepsilon^j z_{s_{ij}} \quad (i = 1, 2, \dots, m).$$

Здесь $x_{s_i} (i = 1, \dots, m)$ — коэффициенты разложения вектора ограничений B исходной задачи по векторам системы $A_{s_1}, \dots, A_{s_m}$; $z_{s_{ij}} (i = 1, 2, \dots, m)$ — коэффициенты разложения вектора условий A_j по векторам системы $A_{s_1}, \dots, A_{s_m}$.

Заметим, что ни один из многочленов $x_{s_i}(\varepsilon)$ не является тождественным нулем. Это следует из того, что в $x_{s_i}(\varepsilon)$ коэффициент $z_{s_i s_i}$ при ε^{s_i} равен 1 ($z_{s_i s_i}$ — коэффициент разложения A_{s_i} по системе векторов $A_{s_1}, \dots, A_{s_m}$ при векторе A_{s_i}). Всякий многочлен степени n , тождественно не равный нулю, имеет не больше n положительных корней. Обозначим минимальный положительный корень многочлена $x_{s_i}(\varepsilon)$ через η_{s_i} . Если $x_{s_i}(\varepsilon)$ не имеет положительных корней, примем $\eta_{s_i} = \infty$. Пусть наименьшее из η_{s_i} ($i = 1, 2, \dots, m$) равно η_s . Величина η_s определяется выбранной системой S линейно независимых векторов $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. Число таких систем конечно. Это значит, что существует $\varepsilon_1 = \min \eta_s$ по всевозможным линейно независимым системам из m векторов условий. Величина ε_1 как минимум из конечного числа положительных чисел положительна ($\varepsilon_1 > 0$).

Пусть теперь $\varepsilon < \varepsilon_1$. Рассмотрим произвольный опорный план ε -задачи. Легко видеть, что он содержит m положительных компонент. Если бы какая-либо его компонента обратилась в нуль, то соответствующий ей многочлен $x_{s_k}(\varepsilon)$ имел бы корнем ε . Но это противоречит допущению, согласно которому все положительные корни любого многочлена $x_{s_i}(\varepsilon)$, в том числе и $x_{s_k}(\varepsilon)$, не менее $\varepsilon_1 > \varepsilon$.

Таким образом, при $\varepsilon < \varepsilon_1$ любой опорный план ε -задачи содержит m положительных компонент, что по определению означает невырожденность задачи. I утверждение доказано.

Докажем теперь II утверждение, согласно которому при достаточно малых ε всякий опорный план ε -задачи порождает опорный план исходной задачи.

Рассмотрим произвольную систему из m линейно независимых векторов условий $A_{s_1}, \dots, A_{s_m}$. Разложим по ней вектор ограничений B и все векторы условий

$$B = \sum_{i=1}^m x_{s_i} A_{s_i}; \quad A_j = \sum_{i=1}^m z_{s_i j} A_{s_i}.$$

Обозначим наибольшее отрицательное (наименьшее по абсолютной величине) значение x_{s_i} через ξ_s . Если все x_{s_i} неотрицательны, полагаем $\xi_s = 0$. Величина ξ_s определяется выбранной системой s линейно независимых векторов $A_{s_1}, \dots, A_{s_m}$. Пусть $\xi = \max \xi_s$ по всем системам из m линейно независимых векторов условий, для которых $\xi_s < 0$. Если все ξ_s равны нулю, то и ξ предполагается равным нулю.

При $\xi = 0$ любая система из m линейно независимых векторов условий является базисом некоторого опорного плана исходной задачи, и утверждение II справедливо для произвольного ε .

Пусть теперь $\xi < 0$. Введем величину

$$\zeta = \max |z_{s_i j}|,$$

где максимум берется по всем j , всем i ($i = 1, 2, \dots, m$) и всем системам S из m линейно независимых векторов условий $A_{s_1}, \dots, A_{s_m}$. В качестве величины $\varepsilon_2 > 0$, фигурирующей во II утверждении, можно выбрать меньшее из двух положительных чисел $-\frac{\xi}{n\zeta}$ и 1

$$\varepsilon_2 = \min \left[-\frac{\xi}{n\zeta}, 1 \right].$$

Покажем, что при любом $\varepsilon < \varepsilon_2$ каждому плану ε -задачи соответствует план исходной задачи. Пусть базисом плана ε -задачи является система векторов условий $A_{s_1}, A_{s_2}, \dots, A_{s_m}$. Тогда компоненты этого плана

$$x_{s_i}(\varepsilon) = x_{s_i} + \sum_{j=1}^n \varepsilon^j z_{s_i j}$$

неотрицательны при любом i . При этом и x_{s_i} неотрицательны для $i = 1, 2, \dots, m$. Пусть это неверно, и при $i = k$ компонента $x_{s_k} < 0$. Тогда

$$\begin{aligned} x_{s_k}(\varepsilon) &= x_{s_k} + \sum_{j=1}^n \varepsilon^j z_{s_k j} \underset{(1)}{\leq} \xi + \sum_{j=1}^n \varepsilon^j \zeta = \\ &= \xi + \zeta \sum_{j=1}^n \varepsilon^j \underset{(2)}{<} \xi + \varepsilon_2 \zeta \sum_{j=0}^{n-1} \varepsilon_2^j \underset{(3)}{\leq} \xi + \varepsilon_2 \zeta n \underset{(4)}{\leq} 0. \end{aligned}$$

Неравенство (1) следует из определения ξ и ζ . Неравенство (2) справедливо для $\varepsilon < \varepsilon_2$. Наконец, неравенства (3) и (4) вытекают из определения ε_2 .

Полученная цепочка соотношений приводит к неравенству

$$x_{s_k}(\varepsilon) < 0,$$

что противоречит допущению о том, что $x_{s_k}(\varepsilon)$ компонента плана ε -задачи.

Чтобы завершить доказательство II утверждения, остается показать, что при $\varepsilon < \varepsilon_2$ оптимальному плану ε -задачи соответствует решение исходной задачи.

Пусть план X_ε с базисом $A_{s_1}, \dots, A_{s_m}$ является оптимальным планом ε -задачи. В соответствии со II формой критерия оптимальности это значит, что

$$\sum_{i=1}^m \mu_{s_i} \cdot a_{s_i j} \geq c_j \quad (j = 1, 2, \dots, n). \quad (3.40)$$

Величины μ_{s_i} определяются только базисом и, следовательно, не зависят от ε . Коэффициенты линейной формы c_j также не зависят от ε . Кроме того, согласно доказанному при $\varepsilon < \varepsilon_2$ базис X_ε является базисом некоторого плана исходной задачи. Для этого плана неравенства (3.40) выполняются при всех j . Следовательно, соответствующий план исходной задачи является ее решением.

II утверждение полностью доказано.

7.3. Приведенные соображения позволяют предложить следующий путь решения задач линейного программирования, не оговаривая их невырожденности.

Вместо исходной задачи рассматривается ε -задача при $\varepsilon < \varepsilon_0$. В качестве $\varepsilon_0 > 0$ выбирается число, не превышающее меньшего из двух чисел ε_1 и ε_2 . В силу I утверждения ($\varepsilon < \varepsilon_1$) рассматриваемая ε -задача невырождена. Поэтому метод последовательного улучшения плана приведет через конечное число шагов к решению задачи. В силу II утверждения ($\varepsilon < \varepsilon_2$) базис полученного решения оказывается вместе с тем базисом оптимального плана исходной задачи. Если решение ε -задачи записано в форме

$$x_i(\varepsilon) = x_i + \sum_{j=1}^n \varepsilon^j z_{ij},$$

то для получения решения исходной задачи достаточно положить ε равным нулю. Если же оптимальный план ε -задачи получен для фиксированного значения ε , то для получения решения исходной задачи следует разложить вектор ограничений по базису оптимального плана ε -задачи.

Заметим, что практически в определении ε_0 нет необходимости. Существование ε_0 важно лишь для обоснования правила решения вырожденных задач. При использовании ε -приема можно считать ε меньшим любого из положительных чисел, с которым его приходится сравнивать в процессе решения задачи.

На основе приведенных соображений можно указать правило удаления вектора из базиса, исключающее возможность зацикливания.

В вырожденной задаче условие

$$\theta = \min_i \frac{x_i}{z_{ik}}, \quad z_{ik} > 0 \quad (3.26)$$

может осуществляться сразу для нескольких значений i . Отсутствие правила, указывающего, какой из векторов A_i нужно при этом выводить из базиса и какой следует оставить, — основное затруднение процедуры. Приведенные выше рассуждения снимают эти трудности. Произведя ε -сдвиг вектора ограничений, перепишем условие (3.26) в виде

$$\theta = \min_{\substack{i \\ z_{ik} > 0}} \frac{x_i(\varepsilon)}{z_{ik}} = \min_{\substack{i \\ z_{ik} > 0}} \left[\frac{x_i}{z_{ik}} + \varepsilon \frac{z_{i1}}{z_{ik}} + \right. \\ \left. + \varepsilon^2 \frac{z_{i2}}{z_{ik}} + \dots + \varepsilon^n \frac{z_{in}}{z_{ik}} \right], \quad (3.26')$$

Здесь $\frac{x_i}{z_{ik}} = \frac{z_{i0}}{z_{ik}}$ — отношения составляющих вектора $A_0 = B$ к соответствующим компонентам вектора A_k , вводимого в базис;

$\frac{z_{i1}}{z_{ik}}$ — отношения составляющих векторов A_1 и A_k и т. д.

Пусть $\min_{z_{ik} > 0} \frac{x_i}{z_{ik}}$ достигается для нескольких значений индекса. Чтобы выбрать из них индекс вектора,

который должен быть изъят из базиса, следует сравнить для них значения $\frac{z_{i1}}{z_{ik}}$ (при $z_{ik} > 0$). Вектор A_r , для которого $\frac{z_{r1}}{z_{ik}}$ принимает наименьшее значение, и является искомым. Может оказаться, что не только $\frac{x_1}{z_{ik}}$, но и $\frac{z_{i1}}{z_{ik}}$ достигает минимума при нескольких (общих для обоих отношений) значениях i . Тогда процесс сравнения отношений компонент очередного вектора условий и соответствующих компонент A_k должен быть продолжен. Таким образом, всегда удастся однозначно выделить вектор A_r , который должен быть удален из базиса и заменен вектором A_k . Приведенное правило выделения вектора A_r , подлежащего выводу из базиса, вытекает из формулы (3.26'). Если исходная задача является вырожденной, то переход от одного базиса к следующему может не сопровождаться увеличением линейной формы. Однако в силу доказанной невырожденности ε -задачи (при достаточно малых ε) зацикливание уже невозможно. В процессе решения задачи исключается возвращение к уже пройденному базису и решение получается в конечное число шагов.

Как мы видели, вырожденный случай не вносит никаких особенностей в правила решения задач по методу последовательного улучшения плана. Необходимо лишь сохранить порядок столбцов в таблицах алгоритма. Любая нумерация векторов A_j может быть принята за исходную, но этот порядок должен сохраниться во всех дальнейших вычислениях. (Строго говоря, порядок векторов необходимо сохранять лишь между двумя последовательными изменениями линейной формы).

Мы уже отмечали, что зацикливание в задачах линейного программирования — редкое явление. Поэтому в ряде случаев целесообразно заменить приведенное здесь правило определения вектора, подлежащего исключению из базиса, более простыми условиями. В частности, можно рекомендовать следующее упрощенное правило. Из базиса исключается тот из «подозрительных» векторов условий, который имеет наименьший номер. Если при этом будет получен цикл, то, начиная от базиса, полученного вторично, следует применять

методику выбора A_r , изложенную в настоящем параграфе. После того, как линейная форма увеличится, целесообразно снова вернуться к упрощенному правилу.

§ 8. ИССЛЕДОВАНИЕ ОБЩИХ ПРОБЛЕМ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ С ПОМОЩЬЮ МЕТОДА ПОСЛЕДОВАТЕЛЬНОГО УЛУЧШЕНИЯ ПЛАНА

Метод улучшения плана, описанный в предыдущих параграфах настоящей главы, позволяет не только решать конкретные задачи. С его помощью можно довольно просто устанавливать многие качественные факты теории линейного программирования. В этом параграфе отмеченное обстоятельство иллюстрируется на примере доказательства двух важных теорем линейного программирования, на которые мы будем неоднократно ссылаться в последующих главах.

Теорема 1. *Для разрешимости задачи линейного программирования необходимо и достаточно выполнение двух условий:*

- а) *задача обладает хотя бы одним планом;*
- б) *линейная форма задачи ограничена на множестве всех ее планов.*

Доказательство. Будем рассматривать произвольную задачу линейного программирования, записанную в канонической форме (3.1), (3.2), (3.3). Без уменьшения общности можно считать все компоненты вектора ограничений $B = (b_1, b_2, \dots, b_m)$ неотрицательными (этого всегда можно добиться за счет умножения соответствующих соотношений системы (3.2) на -1). Необходимость условий теоремы очевидна. Обратимся к доказательству достаточности. Свяжем с задачей (3.1) — (3.3) следующую вспомогательную задачу линейного программирования.

Обратить в максимум линейную форму

$$-\sum_{i=1}^m x_{n+i} \quad (3.41)$$

при соблюдении условий

$$\sum_{j=1}^n a_{ij}x_j + x_{n+i} = b_i, \quad i=1, 2, \dots, m; \quad (3.42)$$

$$x_i \geq 0, \quad i=1, 2, \dots, n+m. \quad (3.43)$$

Нетрудно заметить, что вектор $\bar{X}_0 = (0, 0, \dots, 0, b_1, b_2, \dots, b_m)$ является опорным планом задачи (3.41), (3.42), (3.43). Поэтому при решении вспомогательной задачи методом улучшения плана его можно принять за исходный план. Каждый шаг метода улучшения плана связан с переходом к новому базису, причем возвращение к ранее пройденному базису невозможно (при вырожденной ситуации мы пользуемся правилами, изложенными в предыдущем параграфе). В силу ограниченности числа различных базисов в любой задаче линейного программирования процесс решения складывается лишь из конечного числа шагов. В п. 1.2 было отмечено, что процедура метода улучшения плана завершается либо случаем 1°, либо случаем 2°. Поскольку линейная форма (3.41) ограничена сверху на множестве планов вспомогательной задачи (числом 0), случай 2° исключается. Следовательно, через конечное число шагов мы обязательно столкнемся со случаем 1°, означающим оптимальность построенного опорного плана. Пусть этим планом является

$$\bar{X}_1 = (x_1^{(1)}, x_2^{(1)}, \dots, x_n^{(1)}, x_{n+1}^{(1)}, \dots, x_{m+n}^{(1)}).$$

В соответствии с условием а) теоремы существует хотя бы один план задачи (3.1) — (3.3). Обозначим его через $X = (x_1, \dots, x_n)$. Вектор $\bar{X} = (x_1, x_2, \dots, x_n, 0, 0, \dots, 0)$, очевидно, является планом вспомогательной задачи, связанным с нулевым значением линейной формы (3.41). В силу оптимальности плана $\bar{X}_1 \sum_{i=1}^m x_{n+i}^{(1)} \leq 0$. Откуда, используя условия (3.43), получаем $x_{n+i}^{(1)} = 0$ для $i = 1, 2, \dots, m$. Таким образом, вектор $X_1 = (x_1^{(1)}, \dots, x_n^{(1)})$ — опорный план исследуемой задачи. Применим теперь метод последовательного улучшения плана к задаче (3.1) — (3.3), приняв за исходный только что построенный план $X_1 = (x_1^{(1)}, \dots, x_n^{(1)})$. Условия теоремы б) исключают возможность появления случая 2°. Следовательно, через конечное число шагов очередная итерация метода последовательного улучшения плана завершится случаем 1°, указывающим на оптимальность соответствующего опорного плана рассматриваемой задачи.

Итак, пользуясь условиями а), б), мы установили существование опорного решения задачи (3.1)—(3.3). Теорема доказана полностью.

Сформулируем еще одно общее утверждение теории линейного программирования.

Теорема 2. *Всякая разрешимая задача линейного программирования обладает хотя бы одним опорным решением.*

Доказательство этой теоремы с очевидностью следует из предыдущей теоремы. Действительно, поскольку исследуемая задача линейного программирования разрешима, для нее выполняются условия а), б). В то же время при доказательстве теоремы I мы вывели из этих условий существование опорного решения рассматриваемой задачи. Следовательно, разрешимая задача линейного программирования обязана иметь опорный оптимальный план. Теорема 2 доказана.

Следует заметить, что использование методов линейного программирования*) для качественных выводов о характере линейных задач всегда приводит к конструктивным доказательствам, устанавливающим не только существование, но и приемлемые пути расчета.

*) Для доказательства ряда фактов теории линейного программирования выгоднее использовать другие общие методы, такие как метод последовательного уточнения оценок, метод последовательного сокращения невязок. Подробному изложению этих методов посвящена следующая глава.

ГЛАВА 4

ОБЩИЕ МЕТОДЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ, ОСНОВАННЫЕ НА ПРИНЦИПЕ ДВОЙСТВЕННОСТИ

Настоящая глава посвящена изложению основных идей одного из центральных пунктов линейного программирования — теории двойственности и вытекающих из нее методов решения линейных задач.

Произвольной задаче линейного программирования можно определенным образом сопоставить некоторую другую линейную задачу, называемую обычно двойственной или сопряженной. Теория двойственности обнаруживает тесную связь между обеими задачами, составляющими единую двойственную пару. При количественных и качественных исследованиях в линейном программировании чрезвычайно плодотворным является совместное рассмотрение обеих взаимосопряженных задач. В частности, при решении некоторых задач линейного программирования выгоднее определить вначале оптимальный план двойственной задачи, а затем уже получить искомый ответ. Иногда бывает полезным совместное решение обеих задач двойственной пары. Каждому из этих подходов соответствует особый метод решения общей задачи линейного программирования.

§ 1. ОСНОВЫ ТЕОРИИ ДВОЙСТВЕННОСТИ

1.1. Постановка вопроса. Анализ задач линейного программирования, как правило, удобнее проводить едиными методами, приспособленными к унифицированной

канонической форме. Тем не менее, в ряде случаев рассмотрение задачи в естественной форме, определяемой конкретным приложением, может существенно облегчить ее исследование. В частности, при изучении проблем двойственности целесообразно рассматривать задачи линейного программирования в различной форме записи.

Рассмотрим следующую задачу линейного программирования. Требуется определить вектор $X = (x_1, x_2, \dots, x_n)$, удовлетворяющий ограничениям

$$\sum_{j=1}^n a_{ij}x_j \leq b_i \quad \text{для } i=1, 2, \dots, m, \quad (4.1)$$

$$x_j \geq 0, \quad j=1, 2, \dots, n \quad (4.2)$$

и доставляющий при этом максимум линейной форме

$$\sum_{j=1}^n c_j x_j. \quad (4.3)$$

Прежде всего напомним экономическую интерпретацию задачи (4.1), (4.2), (4.3), которую в дальнейшем будем кратко обозначать римской цифрой I (эта интерпретация изложена в § 8 гл. 1).

Имеется n способов производства некоторого однородного продукта. Количество продукта, получаемое по j -му способу производства в единицу времени, составляет c_j единиц. Использование j -го способа производства в течение единицы времени связано с расходом i -го фактора производства, $i=1, 2, \dots, m$, измеряемым величиной a_{ij} . Допустим, что запасы факторов производства составляют соответственно $b_1, b_2, \dots, b_m$ единиц. Через x_j обозначим время, в течение которого производство ведется по j -му способу, $j=1, 2, \dots, n$.

Задача I является математической формулировкой проблемы составления такого плана использования различных способов производства, который позволяет при наличии определенных ресурсов $(b_1, b_2, \dots, b_m)$ получить максимальное количество однородного продукта. Допустим теперь, что нам необходимо, оставаясь в рамках рассматриваемого производства, оценить каждый из его факторов.

Одни и те же производственные факторы в условиях разных предприятий и районов представляют различную ценность.

Мы здесь будем рассматривать лишь идеализированную замкнутую модель производства, в которой связи с внешним миром строго фиксированы условиями задачи. Условия задачи (ограниченные ресурсы и отработанные способы производства) определяют оценку каждого фактора, внутреннюю для данного производства.

Эта оценка является относительной. Во-первых, такая оценка характеризует лишь лимитированность фактора в условиях анализируемого производства. Во-вторых, оценка производственного фактора измеряется в единицах ценности выпускаемой продукции. Ценность продукции определяется условиями, внешними по отношению к рассматриваемому производству.

Наряду с термином оценка фактора будем пользоваться терминами цена или стоимость фактора как синонимами.

Следует подчеркнуть, что термины оценка, стоимость и цена понимаются здесь только в указанном выше смысле. Примем стоимость единицы однородного продукта за единицу. Через y_i , $i=1, 2, \dots, m$, обозначим цену единицы i -го фактора производства (оценку i -го фактора). Проанализируем j -й способ производства с точки зрения расходов и доходов. Если этот способ использовать в течение единицы времени, то стоимость полученной продукции составит c_j , а стоимость всех затраченных ресурсов окажется равной

$$\sum_{i=1}^m a_{ij} y_i.$$

Отсюда следует, что цены $y_1, y_2, \dots, y_m$ должны удовлетворять неравенству

$$\sum_{i=1}^m a_{ij} y_i \geq c_j$$

(при рациональных оценках факторов стоимость затрат труда, сырья и т. п. не может быть меньше стоимости

окончательного продукта). Кроме того, естественно считать цены $y_1, y_2, \dots, y_m$ неотрицательными числами.

Итак, величины $y_1, \dots, y_m$ обязаны удовлетворять условиям

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n; \quad (4.4)$$

$$y_i \geq 0, \quad i = 1, 2, \dots, m. \quad (4.5)$$

Конечно, существует бесчисленное множество векторов $Y = (y_1, \dots, y_m)$, подчиняющихся ограничениям (4.4), (4.5). (В качестве Y можно, например, взять любой вектор с достаточно большими компонентами.)

Необходимо ввести условия, не допускающие необоснованного завышения оценок факторов производства. Естественным ограничением подобного типа является следующее: вектор оценок должен быть таким, чтобы суммарная оценка ресурсов, которыми располагает производство,

$$\sum_{i=1}^m b_i y_i \quad (4.6)$$

достигла при соблюдении условий (4.4), (4.5) своего минимума. Таким образом, проблема оценки факторов производства формулируется математически как задача линейного программирования, заключающаяся в минимизации формы (4.6) при условиях (4.4), (4.5). Задачу (4.4), (4.5), (4.6), обозначаемую в дальнейшем кратко II, принято называть двойственной задаче I или сопряженной с ней. Нетрудно убедиться, что задачей, сопряженной с II, является исходная задача I. Для этого достаточно превратить II в задачу максимизации вида (4.1) — (4.3) (положить $b'_i = -b_i$ и изменить знаки неравенства в (4.4) на противоположные, введя новые параметры $a'_{ij} = -a_{ij}$) и затем воспользоваться изложенным правилом составления двойственной задачи.

В связи с этим задачи I, II естественно называть двойственной или взаимосопряженной парой. Пару двойственных проблем удобно задавать в виде следующей наглядной табл. 4.1.

Таблица 4.1

		Производительность												
		c_1	c_2	$\cdot$	$\cdot$	$\cdot$	c_j	$\cdot$	$\cdot$	$\cdot$	c_{n-1}	c_n		
План цен	y_1	a_{11}	a_{12}	$\cdot$	$\cdot$	$\cdot$	a_{1j}	$\cdot$	$\cdot$	$\cdot$	$a_{1,n-1}$	a_{1n}	b_1	Ресурсы
	y_2	a_{21}	a_{22}	$\cdot$	$\cdot$	$\cdot$	a_{2j}	$\cdot$	$\cdot$	$\cdot$	$a_{2,n-1}$	a_{2n}	b_2	
	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	
	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	
	y_l	a_{l1}	a_{l2}	$\cdot$	$\cdot$	$\cdot$	a_{lj}	$\cdot$	$\cdot$	$\cdot$	$a_{l,n-1}$	a_{ln}	b_l	
	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	
	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	$\cdot$	
	y_m	a_{m1}	a_{m1}	$\cdot$	$\cdot$	$\cdot$	a_{mj}	$\cdot$	$\cdot$	$\cdot$	$a_{m,n-1}$	a_{mn}	b_m	
		x_1	x_2	$\cdot$	$\cdot$	$\cdot$	x_j	$\cdot$	$\cdot$	$\cdot$	x_{n-1}	x_n		
		План производства												

Пусть $X=(x_1, x_2, \dots, x_n)$ и $Y=(y_1, y_2, \dots, y_m)$ — планы задач I и II соответственно. Покажем, что

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i. \tag{4.7}$$

Пользуясь условиями (4.4), которым обязан удовлетворять вектор Y , имеем

$$\sum_{j=1}^n c_j x_j \leq \sum_{j=1}^n x_j \sum_{i=1}^m a_{ij} y_i = \sum_{i=1}^m y_i \sum_{j=1}^n a_{ij} x_j. \tag{4.8}$$

С другой стороны, из системы неравенств (4.1) следует, что

$$\sum_{i=1}^m b_i y_i \geq \sum_{i=1}^m y_i \sum_{j=1}^n a_{ij} x_j. \tag{4.9}$$

Сравнивая (4.8) и (4.9), получаем искомое неравенство (4.7). Соотношение (4.7) во многих случаях оказы-

ваётся полезным. В частности, из него легко следует условие, гарантирующее оптимальность планов задач I, II. Сформулируем это условие.

Если для некоторых планов $X^* = (x_1^*, \dots, x_n^*)$, $Y^* = (y_1^*, \dots, y_m^*)$ выполняется равенство

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*, \quad (4.10)$$

то вектора X^* и Y^* являются решениями задач I и II соответственно.

Действительно, согласно (4.7) для любого плана $X = (x_1, \dots, x_n)$ задачи I справедливо неравенство

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i^* = \sum_{j=1}^n c_j x_j^*.$$

Пользуясь теперь произвольностью плана $X = (x_1, \dots, x_n)$, приходим к выводу об оптимальности $X^* = (x_1^*, \dots, x_n^*)$. Аналогично проверяется оптимальность плана $Y^* = (y_1^*, \dots, y_m^*)$.

Таким образом, если при некотором плане цен стоимость произведенной продукции совпадает со стоимостью имеющихся в запасе ресурсов, то как план цен, так и соответствующий план производства являются оптимальными.

В дальнейшем мы убедимся, что для оптимальности планов задач I, II условие (4.10) является не только достаточным, но и необходимым.

1.2. Теоремы двойственности. Взаимосопряженные задачи линейного программирования обладают рядом интересных свойств, связывающих их воедино. В настоящем пункте мы сформулируем две основные теоремы двойственности и выведем из них несколько полезных следствий. Доказательства этих теорем, опирающиеся на некоторые факты из теории выпуклых множеств, выделены в § 2 гл. 6.

Первая теорема двойственности. Если одна из пары двойственных задач I, II имеет решение, то другая задача также разрешима. При этом для любых оптималь-

ных планов задач I, II $X = (x_1, \dots, x_n)$, $Y = (y_1, \dots, y_m)$ справедливо соотношение (4.10):

$$\boxed{\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i y_i.}$$

В качестве следствия из первой теоремы двойственности получаем условие разрешимости задачи линейного программирования.

Следствие 1. Для разрешимости задачи I (или II) необходимо и достаточно, чтобы множество планов каждой из задач двойственной пары I, II было непустым, т. е. чтобы каждая из задач имела по крайней мере один план.

Достаточность. Пусть $Y = (y_1, \dots, y_m)$ — произвольный план задачи II. Тогда по неравенству (4.7) для любого плана задачи I $X = (x_1, \dots, x_n)$ имеет место условие

$$\sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i.$$

Итак, множество планов задачи I непусто, и линейная форма (4.3) на нем ограничена сверху. Поэтому (теорема I гл. 3) задача I обладает оптимальным планом.

Необходимость. Пусть задача I разрешима. Тогда по первой теореме двойственности разрешима также и задача II. Следовательно, множество ее планов непусто.

При исследовании пары двойственных задач можно столкнуться с одной из трех взаимоисключающих возможностей:

- а) обе задачи обладают планами;
- б) план имеется лишь у одной из задач;
- в) для каждой задачи двойственной пары I, II множество планов пусто.

Приведем соответствующие примеры.

- а) Задача I' состоит в максимизации формы $5x_1 + x_2$ при условиях

$$\begin{aligned} x_1 + 2x_2 &\leq 4, \\ 2x_1 + 3x_2 &\leq 3, \\ x_1 &\geq 0, \quad x_2 \geq 0. \end{aligned}$$

Сопряженная задача Π' , составленная по общему правилу, имеет вид

$$\begin{aligned} 4y_1 + 3y_2 &= \min; \\ y_1 + 2y_2 &\geq 5, \\ 2y_1 + 3y_2 &\geq 1, \\ y_1 &\geq 0, \quad y_2 \geq 0. \end{aligned}$$

Очевидно, что множество планов каждой из этих задач непусто

$(0,0)$ — план задачи I' ;

$(1,2)$ — план задачи Π' .

б) Задача I'' имеет вид

$$\begin{aligned} 5x_1 + x_2 &= \max; \\ x_1 - 2x_2 &\leq 2, \\ x_1 - 3x_2 &\leq 3, \\ x_1 &\geq 0, \quad x_2 \geq 0. \end{aligned}$$

Задача Π'' :

$$\begin{aligned} 2y_1 + 3y_2 &= \min; \\ y_1 + y_2 &\geq 5, \\ -2y_1 - 3y_2 &\geq 1, \\ y_1 &\geq 0, \quad y_2 \geq 0. \end{aligned}$$

Вектор $(0, 0)$ является планом задачи I'' .

Множество планов задачи Π'' — пустое множество. В самом деле, умножая первое условие задачи Π'' на 2 и складывая со вторым, имеем $-y_2 \geq 11$, т. е. $y_2 \leq -11$ что противоречит третьему условию этой задачи.

в) Задача I''' :

$$\begin{aligned} 5x_1 + x_2 &= \max; \\ x_1 - x_2 &\leq 1, \\ -x_1 + x_2 &\leq -2, \\ x_1 &\geq 0, \quad x_2 \geq 0. \end{aligned}$$

Задача Π''' :

$$\begin{aligned} y_1 - 2y_2 &= \min; \\ y_1 - y_2 &\geq 5, \\ -y_1 + y_2 &\geq 1, \\ y_1 &\geq 0, \quad y_2 \geq 0. \end{aligned}$$

Предоставим читателю проверить, что множество планов каждой из приведенных задач пусто.

Из следствия 1 вытекает, что в случае а) обе задачи двойственной пары разрешимы. Что касается случая б), то линейная форма задачи, множество планов которой непусто, заведомо неограничена (сверху для задачи I, снизу для задачи II). Действительно, если это не так, одна из задач двойственной пары обладает оптимальным планом. Тогда по первой теореме двойственности другая задача также разрешима, что противоречит условию б).

Из следствия 1, в частности, вытекает разрешимость довольно широкого класса линейных задач типа I.

Допустим, что величины b_i , $i=1, 2, \dots, m$ неотрицательны, а величины a_{ij} для некоторого i и $j=1, 2, \dots, n$, строго положительны. Тогда задача I (а значит и задача II) разрешима.

Для доказательства заметим, что вектор $(0, 0, \dots, 0)$ является планом задачи I.

С другой стороны, из положительности элементов i -й строки матрицы $A = \|a_{ij}\|_{m,n}$ вытекает наличие планов у задачи II.

Действительно, при этих условиях вектор $\overbrace{(0, 0, \dots, 0, y, 0, 0, \dots, 0)}^i$ при достаточно большом значении y удовлетворяет всем условиям системы (4.4) и является, следовательно, планом задачи II. Итак, обе задачи двойственной пары обладают планами. Тогда по следствию I они разрешимы.

В качестве второго следствия из первой теоремы двойственности получим необходимое и достаточное условие оптимальности планов задач I, II.

Следствие 2. Для оптимальности планов $X^* = (x_1^*, \dots, x_n^*)$, $Y^* = (y_1^*, \dots, y_m^*)$ задач I, II необходимо и достаточно выполнение условия (4.10)

$$\sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*.$$

Достаточность этого условия уже установлена.

Необходимость составляет содержание 2-й части первой теоремы двойственности.

Для формулировки второй теоремы двойственности необходимо ввести несколько новых понятий.

При любом $i(j)$ совокупность i -го (j -го) неравенства системы (4.1) (соответственно (4.2)): $\sum_{j=1}^n a_{ij}x_j \leq b_i$ ($x_j \geq 0$) и i -го (j -го) неравенства системы (4.5) ((4.4)): $y_i \geq 0 \left(\sum_{i=1}^m a_{ij}y_i \geq c_j \right)$ назовем парой двойственных условий задач I, II.

Если обратиться к таблице 4.1, определяющей задачи I, II (стр. 221), то пара двойственных условий $\sum_{j=1}^n a_{ij}x_j \leq b_i$ $y_i \geq 0$ соответствует i -й строке таблицы,

пара двойственных условий $\sum_{i=1}^m a_{ij}y_i \geq c_j$, $x_j \geq 0$ — ее j -му столбцу. Назовем неравенство, связывающее переменные задачи I (соответственно задачи II), закрепленным условием, если оно выполняется как равенство при любом оптимальном плане задачи I (II). Другими словами, если соотношение $\sum_{j=1}^n a_{ij}x_j = b_i$ или $x_j = 0 \left(\sum_{i=1}^m a_{ij}y_i = c_j \right.$ или $y_i = 0 \left. \right)$ имеет место для всех решений задачи I (II), то соответствующее условие системы (4.1) или (4.2) ((4.4) или (4.5)) является закрепленным.

Неравенство, связывающее переменные задачи I (II), назовем свободным условием, если хотя бы для одного оптимального плана задачи I (II) оно выполняется как строгое неравенство. Таким образом, если существует такой оптимальный план задачи I $X^* = (x_1^*, x_2^*, \dots, x_n^*)$, что $\sum_{j=1}^n a_{ij}x_j^* < b_i$ или $x_j^* > 0$, то соответствующее условие задачи I является свободным. Аналогично, если для некоторого решения задачи II $Y^* = (y_1^*, y_2^*, \dots, y_m^*)$ $\sum_{i=1}^m a_{ij}y_i^* > c_j$ или $y_i^* > 0$, то j -е условие системы (4.4) и, соответственно, i -е условие (4.5), считается свободным.

Вторая теорема двойственности устанавливает связь между парами двойственных условий задач I, II.

Вторая теорема двойственности. Если взаимосопреженные задачи I, II разрешимы, то в каждой паре их двойственных условий одно — свободное, а другое — закрепленное.

Таким образом, если рассмотреть i -ю пару двойственных условий

$$\sum_{j=1}^n a_{ij}x_j \leq b_i (*), \quad y_i \geq 0,$$

соответствующую i -й строке таблицы задач I, II, то либо (*) на всех оптимальных планах задачи I обращается в равенство, причем i -я компонента хотя бы одного решения задачи II положительна, либо все оптимальные планы задачи II имеют нулевую i -ю компоненту, а неравенство (*) является строгим для некоторого решения задачи I. То же верно и для любой пары двойственных условий, соответствующей столбцу таблицы задач I, II.

Как первая, так и вторая теоремы двойственности будут неоднократно использоваться в дальнейшем изложении. Если переформулировать теоремы двойственности в терминах производства однородного продукта, то их утверждения приобретают довольно наглядный и прозрачный смысл.

По первой теореме двойственности оптимальный план производства существует в том и только в том случае, если может быть определен оптимальный план цен, компоненты которого естественно назвать ценами соответствующих факторов производства. Допустим, что производство обладает оптимальным планом и, следовательно, все его факторы могут быть оценены. Следует заметить, что в реальных случаях это всегда так. Вообще говоря, как оптимальный план производства, так и оценки его факторов определяются неоднозначно. По второй части первой теоремы двойственности при любых оценках производственных факторов (составляющих оптимальный план цен) стоимость продукта, полученного реализацией оптимального плана производства, совпадает с суммарной оценкой имеющихся в запасе ресурсов. Таким образом, характеристическое свойство

оптимального плана производства состоит в совпадении по стоимости результата производства и его затрат.

Перейдем к анализу второй теоремы двойственности. Положим, что производство обладает единственным оптимальным планом и оценки его факторов, составляющие оптимальный план цен, определяются однозначно (задачи I, II имеют единственные решения). Пусть в оптимальном плане производства i -й фактор расходуется не полностью. Тогда по второй теореме двойственности его цена равна нулю. Наоборот, если оценка некоторого фактора производства равна нулю, то в оптимальном плане этот фактор полностью не используется. Такое положение довольно естественно. В самом деле, фактор, запасы которого превышают потребности в нем, не представляет ценности для производства: некоторое сокращение количества такого фактора не уменьшит возможностей производства. Поэтому оценку подобного фактора с точки зрения данного производства естественно принять равной нулю. Допустим теперь, что излишек рассматриваемого фактора изъят. В этом случае он приобретает для производства известную ценность: дальнейшее уменьшение его запасов ведет теперь к сокращению конечной продукции. Оценка фактора становится положительной.

Обратимся к другой группе двойственных условий. Согласно второй теореме двойственности данный способ производства используется в оптимальном плане в том и только в том случае, если при его реализации стоимости полученного продукта и затраченных ресурсов совпадают. Интуитивно это утверждение совершенно очевидно: если некоторый способ производства связан с превышением расходов над доходами, то его использование не имеет смысла.

1.3. Задачи линейного программирования со смешанными условиями. В задачах I, II, которые до сих пор нами рассматривались в настоящем параграфе, все ограничения имеют вид неравенств. При этом каждое переменное предполагается неотрицательным. Подобные задачи естественно называть задачами с **о д н о т и п н ы м и** **у с л о в и я м и**. Обратимся теперь к более общему классу задач линейного программирования, у которых часть ограничений суть строгие равенства и некоторые из переменных могут быть отрицательными. Представи-

телей этого класса будем называть задачами со смешанными условиями. Произвольная задача такого типа формулируется следующим образом.

Требуется определить вектор $X = (x_1, x_2, \dots, x_{n-1}, x_n)$, удовлетворяющий условиям

$$\left. \begin{aligned} \sum_{j=1}^n a_{ij}x_j &\leq b_i, \quad i = 1, 2, \dots, m_1; \\ \sum_{j=1}^n a_{ij}x_j &= b_i, \quad i = m_1 + 1, \dots, m; \end{aligned} \right\} \quad (4.11)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n_1 \leq n \quad (4.12)$$

и обращающий при этом в максимум линейную форму

$$\sum_{j=1}^n c_j x_j. \quad (4.13)$$

То, что именно первые m_1 условий (4.11) имеют вид неравенств и ограничения (4.12) относятся к первым n_1 переменным, несколько не уменьшает общности постановки, поскольку любую задачу со смешанными условиями можно за счет перенумерации ограничений и переменных свести к виду (4.11), (4.12), (4.13). Введенную задачу будем в дальнейшем обозначать цифрой III.

Задача, сопряженная проблеме III, формулируется так.

Определить вектор $Y = (y_1, y_2, \dots, y_{m-1}, y_m)$, который при соблюдении ограничений

$$\sum_{i=1}^m a_{ij}y_i \geq c_j, \quad j = 1, 2, \dots, n_1; \quad (4.14)$$

$$\sum_{i=1}^m a_{ij}y_i = c_j, \quad j = n_1 + 1, \dots, n;$$

$$y_i \geq 0, \quad i = 1, 2, \dots, m_1 \quad (4.15)$$

обращает линейную форму

$$\sum_{i=1}^m b_i y_i \quad (4.16)$$

в минимум.

Таким образом, задача (4.14), (4.15), (4.16), обозначаемая в дальнейшем цифрой IV, составляется в соот-

ветствии со следующими правилами. Если переменная x_j в задаче III предполагалась неотрицательной, j -е условие системы (4.14) имеет вид неравенства. Если же на x_j подобного ограничения не накладывалось, j -е условие является равенством. Аналогичные связи имеются между условиями (4.11), (4.15): при i -м условии типа неравенства $y_i \geq 0$, при i -м условии типа равенства y_i может принимать как положительные, так и отрицательные значения. Очевидно, что задача III двойственна задаче IV. Поэтому проблемы III, IV имеет смысл называть парой взаимосопряженных (двойственных) задач, не выделяя среди них исходной. Для взаимосопряженных задач со смешанными условиями справедливы обе теоремы двойственности. Чтобы установить это, сведем двойственную пару III, IV к двойственной паре задач типа I, II (с однотипными условиями).

Свяжем с каждым n -мерным вектором $X = (x_1, \dots, x_n)$ $(n + n_2)$ -мерный вектор $X' = (x'_1, x'_2, \dots, x'_{n+n_2})$ ($n_2 = n - n_1$), полагая

$$x'_j = \begin{cases} x_j & \text{для } j = 1, 2, \dots, n_1, \\ \max(0, x_j) & \text{для } j = n_1 + 1, \dots, n, \\ \max(0, -x_{j-n_2}) & \text{для } j = n + 1, \dots, n + n_2. \end{cases} \quad (4.17)$$

Аналогично каждому вектору $\bar{Y} = (y_1, y_2, \dots, y_m)$ поставим в соответствие вектор $Y' = (y'_1, y'_2, \dots, y'_{m+m_2})$ ($m_2 = m - m_1$), координаты которого определяются формулами

$$y'_i = \begin{cases} y_i & \text{для } i = 1, 2, \dots, m_1, \\ \max(0, y_i) & \text{для } i = m_1 + 1, \dots, m, \\ \max(0, -y_{i-m_2}) & \text{для } i = m + 1, \dots, m + m_2. \end{cases} \quad (4.18)$$

Введем в рассмотрение пару двойственных задач с однотипными условиями и переменными $x'_1, \dots, x'_{n+n_2}$; $y'_1, \dots, y'_{m+m_2}$.

1. Максимизировать линейную форму

$$\sum_{j=1}^n c_j x'_j - \sum_{j=n+1}^{n+n_2} c_{j-n_2} x'_j \quad (4.19)$$

при условиях:

$$\sum_{j=1}^n a_{ij}x_j' - \sum_{j=n+1}^{n+n_2} a_{ij-n_2}x_j' \leq b_i, \quad i=1, 2, \dots, m; \quad (4.20)$$

$$-\sum_{j=1}^n a_{ij}x_j' + \sum_{j=n+1}^{n+n_2} a_{ij-n_2}x_j' \leq -b_i, \quad (4.20')$$

$$i=m_1+1, \dots, m;$$

$$x_j' \geq 0, \quad j=1, 2, \dots, n+n_2. \quad (4.21)$$

2. Минимизировать линейную форму

$$\sum_{i=1}^m b_i y_i' - \sum_{i=m+1}^{m+m_2} b_{i-m_2} y_i' \quad (4.22)$$

при условиях:

$$\sum_{i=1}^m a_{ij} y_i' - \sum_{i=m+1}^{m+m_2} a_{i-m_2,j} y_i' \geq c_j, \quad j=1, 2, \dots, n; \quad (4.23)$$

$$-\sum_{i=1}^m a_{ij} y_i' + \sum_{i=m+1}^{m+m_2} a_{i-m_2,j} y_i' \geq -c_j, \quad (4.23')$$

$$j=n_1+1, \dots, n;$$

$$y_i' \geq 0, \quad i=1, 2, \dots, m+m_2. \quad (4.24)$$

Обозначив задачи (4.19), (4.20), (4.20'), (4.21), (4.22), (4.23), (4.23'), (4.24) через III', IV' соответственно, установим их эквивалентность двойственной паре III, IV. Для этого прежде всего заметим, что согласно формулам (4.17)

$$x_j = x_j' - x_{j+n_2}', \quad j=n_1+1, \dots, n.$$

В самом деле, при $x_j \geq 0$

$$x_j' - x_{j+n_2}' = \max(0, x_j) - \max(0, -x_j) = x_j - 0 = x_j.$$

При $x_j < 0$

$$x_j' - x_{j+n_2}' = 0 - (-x_j) = x_j.$$

Таким образом, формулы (4.17) можно обратить:

$$x_j = \begin{cases} x_j', & j=1, 2, \dots, n_1, \\ x_j' - x_{j+n_2}', & j=n_1+1, \dots, n, \end{cases} \quad (4.25)$$

Пусть $X' = (x_1', \dots, x_{n+n_2}') — план задачи III'.$ С помощью соотношений (4.25) легко получить, что соответствующий вектор $X = (x_1, \dots, x_n)$ является планом задачи II.

Действительно,

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j' - \sum_{j=n+1}^{n+n_2} a_{ij-n_2} x_j' &= \sum_{j=1}^{n_1} a_{ij} x_j' + \\ &+ \sum_{j=n_1+1}^n a_{ij} (x_j' - x_{j+n_2}') = \sum_{j=1}^{n_1} a_{ij} x_j + \\ &+ \sum_{j=n_1+1}^n a_{ij} x_j = \sum_{j=1}^n a_{ij} x_j. \end{aligned}$$

Используя далее условия (4.20), (4.20') задачи III', получаем

$$\begin{aligned} \sum_{j=1}^n a_{ij} x_j &\leq b_i, \quad i = 1, 2, \dots, m_1, \\ \sum_{j=1}^n a_{ij} x_j &= b_i, \quad i = m_1 + 1, \dots, m. \end{aligned}$$

Кроме того, при $j = 1, 2, \dots, n_1$ $x_j = x_j' \geq 0$.

Итак, $X = (x_1, \dots, x_n)$ удовлетворяет ограничениям задачи III и, следовательно, составляет ее план. Подобным же образом проверяется, что вектор $X' = (x_1', x_2', \dots, x_{n+n_2}')$, составленный согласно формулам (4.17) по плану $X = (x_1, \dots, x_n)$ задачи III, является планом задачи III'.

Пусть $X = (x_1, \dots, x_n)$ и $X' = (x_1', \dots, x_{n+n_2}')$ — произвольные векторы, связанные между собой соотношениями (4.17), (4.25). Покажем, что подстановки X и X' в линейные формы (4.13), (4.19) соответственно приводят к тождественным результатам

$$\begin{aligned} \sum_{j=1}^n c_j x_j &= \sum_{j=1}^{n_1} c_j x_j + \sum_{j=n_1+1}^n c_j x_j = \\ &= \sum_{j=1}^{n_1} c_j x_j' + \sum_{j=n_1+1}^n c_j (x_j' - x_{j+n_2}') = \\ &= \sum_{j=1}^n c_j x_j' - \sum_{j=n+1}^{n+n_2} c_{j-n_2} x_j'. \end{aligned} \quad (4.26)$$

Из равенства (4.26) вытекает, что оптимальные планы задач III, III' также связаны соотношениями (4.17), (4.25).

Этим доказана эквивалентность задач III, III', соответствие между переменными которых дается формулами (4.17), (4.25).

Установление эквивалентности задач IV, IV' ничем не отличается от только что проведенного и поэтому может быть опущено.

Теперь мы в состоянии перенести утверждения теорем двойственности на задачи со смешанными условиями.

Формулировка первой теоремы двойственности для общего случая ничем не отличается от ее формулировки применительно к задачам с однотипными условиями.

Что касается второй теоремы двойственности, то ее утверждения относятся лишь к условиям типа неравенств и тем из переменных, которые предполагаются неотрицательными. В задачах III, IV такими условиями являются первые m_1 и n_1 из систем (4.11) и (4.14) соответственно, а неотрицательными переменными — $x_1, x_2, \dots, x_{n_1}; y_1, y_2, \dots, y_{m_1}$. Доказательства обеих теорем двойственности в общем случае легко проводятся по следующей схеме: от задачи со смешанными условиями (III) переходят к задаче с однотипными условиями (III'), затем, используя соответствующую теорему двойственности (III' $\rightarrow$ IV'), возвращаются к сопряженной задаче со смешанными условиями (IV).

Проиллюстрируем эту схему на примере первой теоремы двойственности.

Пусть задача III разрешима и $X = (x_1, \dots, x_n)$ — один из ее оптимальных планов. Рассмотрим задачу III'. Согласно доказанному вектор $X' = (x'_1, \dots, x'_{n+n_2})$, связанный с X соотношениями (4.17), является решением задачи III'. Откуда в соответствии с первой теоремой двойственности для задач с однотипными условиями задача IV' разрешима и для всякого ее оптимального плана $Y' = (y'_1, \dots, y'_{m+m_2})$ имеет место следующее равенство:

$$\sum_{j=1}^n c_j x'_j - \sum_{j=n+1}^{n+n_2} c_{j-n_2} x'_j = \sum_{i=1}^m b_i y'_i - \sum_{i=m+1}^{m+m_2} b_{i-m_2} y'_i.$$

Но оптимальному плану задачи IV' соответствует [по формулам (4.18)] оптимальный же план задачи IV (значит она разрешима). Обозначим его через $Y = (y_1, \dots, y_m)$. Пользуясь далее равенствами

$$\sum_{j=1}^n c_j x_j' - \sum_{j=n+1}^{n+n_2} c_{j-n_2} x_j' = \sum_{j=1}^n c_j x_j,$$

$$\sum_{i=1}^m b_i y_i' - \sum_{i=m+1}^{m+m_2} b_{i-m_2} y_i' = \sum_{i=1}^m b_i y_i,$$

приходим к искомому соотношению

$$\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i y_i,$$

справедливому для произвольных оптимальных планов задач III, IV. Тем самым первая теорема двойственности для задач со смешанными условиями доказана полностью.

1.4. Задачи линейного программирования в канонической форме. Рассмотрим задачу линейного программирования, записанную в канонической форме. Требуется найти вектор $X = (x_1, x_2, \dots, x_n)$, на котором линейная форма

$$\sum_{j=1}^n c_j x_j \tag{4.27}$$

достигает максимума при соблюдении условий:

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m; \tag{4.28}$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \tag{4.29}$$

Эта задача, обозначаемая в дальнейшем цифрой V, является частным случаем задачи III при $m_1 = 0$, $n_1 = n$. В соответствии с общим правилом составления двойственных задач задача VI, сопряженная с V, формулируется следующим образом.

Требуется обратить в минимум линейную форму

$$\sum_{i=1}^m b_i y_i \tag{4.30}$$

при наличии ограничений

$$\sum_{i=1}^m a_{ij}y_i \geq c_j, \quad j=1, 2, \dots, n. \quad (4.31)$$

Таким образом, переменные двойственной задачи VI не обязаны быть неотрицательными.

Мы покажем сейчас, что компоненты вектора $Y^* = (y_1^*, \dots, y_m^*)$, являющегося оптимальным планом задачи VI, могут интерпретироваться как оценки влияния различных условий системы (4.28) на величину максимума задачи V. Сформулируем наше утверждение точнее.

Пусть $M(b_1, \dots, b_m)$ обозначает максимум линейной формы (4.27) при условиях (4.28), (4.29). В таком случае

$$y_i^* = \frac{\partial M(b_1, \dots, b_m)}{\partial b_i}, \quad i=1, 2, \dots, m. \quad (4.32)$$

В дальнейших рассуждениях задача V предполагается невырожденной.

Пусть $X^* = (x_1^*, \dots, x_n^*)$ — опорное решение задачи V. Без ограничения общности можно считать отличными от нуля первые m компонент вектора X^* . Тогда векторы условий $A_j = (a_{1j}, a_{2j}, \dots, a_{mj})$, $j=1, 2, \dots, m$, линейно независимы. Пользуясь этим обстоятельством, можно любой m -мерный вектор $B' = (b'_1, \dots, b'_m)$ представить в виде линейной комбинации векторов $A_1, A_2, \dots, A_m$

$$B' = \sum_{j=1}^m x'_j A_j. \quad (4.33)$$

При этом для определения коэффициентов разложения (4.33) получаем систему m линейных уравнений с определителем, отличным от нуля.

Решая эту систему с помощью известных формул Крамера, нетрудно установить следующие соотношения:

$$x'_j = \sum_{i=1}^m \bar{a}_{ij} b'_i, \quad j=1, 2, \dots, m, \quad (*) \quad (4.34)$$

*) Матрица $\|\bar{a}_{ij}\|$ является обратной для неособенной матрицы $\|a_{ij}\|$.

где величины $\bar{a}_{ij}$ не зависят от компонент вектора $B' = (b_1', \dots, b_m')$.

$$\text{Положим } \bar{a} = \max_{1 \leq i \leq m} \sum_{j=1}^m \bar{a}_{ij}, \quad x = \min_{1 \leq i \leq m} x_i^*.$$

Рассмотрим задачу линейного программирования V' , образующуюся из V путем замены вектора ограничений $B = (b_1, b_2, \dots, b_m)$ на $B' = (b_1', b_2', \dots, b_m')$. Проверим, что при $|b_i - b_i'| \leq \frac{x}{\bar{a}}, i = 1, 2, \dots, m$, n -мерный вектор $X' = (x_1', \dots, x_m', 0, \dots, 0)$, где $x_1', \dots, x_m'$ определяются системой соотношений (4.34), является решением задачи V' .

Покажем вначале, что X' — план задачи V' . По формулам (4.34)

$$x_i' - x_i^* = \sum_{j=1}^m \bar{a}_{ij} (b_j' - b_j), \quad i = 1, 2, \dots, m.$$

Откуда $|x_i' - x_i^*| \leq \max_{1 \leq i \leq m} |b_i' - b_i| \cdot \bar{a} \leq x$ и, следовательно,

$$x_i' \geq x_i^* - x = x_i^* - \min_{1 \leq i \leq m} x_i^* \geq 0.$$

Что касается условий (4.28), в которых вектор B заменен на B' , то они удовлетворяются вектором X' в соответствии с определением величин $x_1', \dots, x_m'$.

Таким образом, $X' = (x_1', \dots, x_m', 0, \dots, 0)$ является планом задачи V' .

По второй теореме двойственности оптимальный план $Y^* = (y_1^*, \dots, y_m^*)$ задачи VI связан с решением задачи V $X^* = (x_1^*, \dots, x_m^*, 0, 0, \dots, 0)$ соотношениями:

$$\sum_{i=1}^m a_{ij} y_i^* = c_j, \quad j = 1, 2, \dots, m,$$

$$\sum_{i=1}^m a_{ij} y_i^* \geq c_j, \quad j = m+1, \dots, n.$$

Поэтому

$$\begin{aligned}\sum_{j=1}^n c_j x_j' &= \sum_{j=1}^m c_j x_j' = \sum_{j=1}^m \left(\sum_{i=1}^m a_{ij} y_i^* \right) x_j' = \\ &= \sum_{i=1}^m y_i^* \left(\sum_{j=1}^m a_{ij} x_j' \right) = \sum_{i=1}^m b_i' y_i^*. \quad (4.35)\end{aligned}$$

Но Y^* является планом задачи VI' , сопряженной с V' (задача VI' получается из VI путем замены коэффициентов линейной формы (4.30) величинами $b_1', b_2', \dots, b_m'$). Поэтому (следствие 2 первой теоремы двойственности) $X' = (x_1', x_2', \dots, x_m', 0, \dots, 0)$ — решение задачи V' . Таким образом, равенство (4.35) может быть переписано в виде

$$M(b_1', \dots, b_m') = \sum_{i=1}^m b_i' y_i^* \quad (4.36)$$

для любых величин $b_1', \dots, b_m'$, удовлетворяющих неравенствам

$$|b_i - b_i'| \leq \delta = \frac{\epsilon}{a} \quad (\delta > 0).$$

Искомые соотношения (4.32) можно получить теперь путем непосредственного дифференцирования обеих частей (4.36).

Итак, рост b_i приводит к увеличению или к уменьшению $M(b_1, \dots, b_m)$ в зависимости от того, положительной или отрицательной оценкой y_i^* обладает i -е условие системы (4.28). При этом скорость изменения $M(b_1, \dots, b_m)$ определяется модулем соответствующей оценки.

1.5. Критерии оптимальности. С помощью теорем двойственности можно получить условия, гарантирующие оптимальность плана задачи линейного программирования.

Обратимся вначале к задаче типа V . Назовем, следуя Л. В. Канторовичу [6], величины $\lambda_1, \lambda_2, \dots, \lambda_m$ разрешающими множителями, если для некоторого

плана $X = (x_1, \dots, x_n)$ задачи V выполняются следующие условия:

$$\sum_{i=1}^m a_{ij} \lambda_i \geq c_j, \quad j = 1, 2, \dots, n, \quad (4.37)$$

причем в случае $x_j > 0$ j -е неравенство системы (4.37) имеет вид строгого равенства

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j \quad \text{при } x_j > 0. \quad (4.38)$$

Вектор $\Lambda = (\lambda_1, \dots, \lambda_m)$, компонентами которого являются разрешающие множители, будем называть **р а з р е ш а ю щ и м**.

Критерии оптимальности плана задачи V^* . Для оптимальности плана $X = (x_1, \dots, x_n)$ задачи V необходимо и достаточно существование разрешающего вектора $\Lambda = (\lambda_1, \dots, \lambda_m)$, связанного с X соотношениями (4.38).

Достаточность. Допустим, что разрешающий вектор $\Lambda = (\lambda_1, \dots, \lambda_m)$ существует.

Если положительные компоненты плана X имеют номера $j_1, j_2, \dots, j_s$, то

$$\sum_{j=1}^n c_j x_j = \sum_{k=1}^s c_{j_k} x_{j_k}.$$

Используя равенства (4.38), которые по условию справедливы для $j = j_1, j_2, \dots, j_s$, получаем

$$\begin{aligned} \sum_{j=1}^n c_j x_j &= \sum_{k=1}^s c_{j_k} x_{j_k} = \sum_{k=1}^s x_{j_k} \left(\sum_{i=1}^m a_{ij_k} \lambda_i \right) = \\ &= \sum_{j=1}^n x_j \left(\sum_{i=1}^m a_{ij} \lambda_i \right) = \sum_{i=1}^m \lambda_i \left(\sum_{j=1}^n a_{ij} x_j \right). \end{aligned}$$

Но $X = (x_1, \dots, x_n)$ — план задачи V . Поэтому $\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m$. Следовательно,

$$\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i \lambda_i.$$

* Впервые подобные критерии были предложены Л. В. Канторовичем [6].

Замечая далее, что согласно (4.37) разрешающий вектор является планом задачи VI, сопряженной с V, и учитывая следствие 2 первой теоремы двойственности, убеждаемся в оптимальности плана

$$X = (x_1, \dots, x_n).$$

Необходимость. Пусть задача V разрешима и $X = (x_1, \dots, x_n)$ — ее оптимальный план.

Установим существование разрешающего вектора, связанного с X системой равенств (4.38). По первой теореме двойственности сопряженная с V задача VI разрешима и, следовательно, обладает хотя бы одним оптимальным планом $\Lambda = (\lambda_1, \dots, \lambda_m)$. Проверим, что $\Lambda = (\lambda_1, \dots, \lambda_m)$ — искомый разрешающий вектор. Поскольку $\Lambda = (\lambda_1, \dots, \lambda_m)$ — план задачи VI, то

$$\sum_{i=1}^m a_{ij} \lambda_i \geq c_j, \quad j = 1, 2, \dots, n.$$

Далее, если $x_j > 0$, j-е условие системы (4.29) является свободным. Тогда в соответствии со второй теоремой двойственности двойственное условие системы (4.31) — закрепленное, и поэтому

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j.$$

Необходимость полностью доказана.

Установленным критерием оптимальности (его достаточностью) мы уже пользовались при описании II-го алгоритма метода последовательного улучшения плана. (Заметим, что в главе 3 разрешающий вектор и приближающие его векторы обозначались буквой μ .)

Понятие разрешающего вектора связывалось до сих пор с конкретным планом задачи V (этот план, естественно, должен быть оптимальным). Однако задача линейного программирования V может, вообще говоря, обладать бесчисленным множеством решений.

Будем называть разрешающий вектор, связанный условиями (4.38) с некоторым оптимальным планом задачи V, разрешающим вектором этого плана.

Спрашивается, существует ли зависимость между разрешающим вектором некоторого оптимального плана задачи V и произвольным ее решением? Ответом на этот вопрос служит следующее утверждение.

Теорема о разрешающих векторах. Разрешающий вектор некоторого оптимального плана задачи V является разрешающим и для любого другого решения этой задачи.

Доказательство. Пусть $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — разрешающий вектор плана $X = (x_1, \dots, x_n)$ задачи V . В процессе доказательства достаточности критерия оптимальности мы установили равенство

$$\sum_{j=1}^n c_j x_j = \sum_{i=1}^m b_i \lambda_i. \quad (4.39)$$

Заметим, что согласно условиям (4.37) $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — план задачи VI .

В таком случае равенство (4.39) приводит к выводу об оптимальности этого плана. Итак, разрешающий вектор $\Lambda = (\lambda_1, \dots, \lambda_m)$ плана X является решением задачи VI .

Рассмотрим теперь произвольный оптимальный план задачи V $X' = (x'_1, \dots, x'_n)$ и покажем, что Λ — его разрешающий вектор. Пусть $x'_j > 0$.

Тогда по определению j -е условие системы (4.29) является свободным. Следовательно (см. вторую теорему двойственности), двойственное ему условие задачи VI [j -е условие системы (4.31)] должно быть закрепленным, т. е. любой оптимальный план задачи VI удовлетворяет ему как равенству. В частности, подставляя в это условие оптимальный план задачи VI $\Lambda = (\lambda_1, \dots, \lambda_m)$, получаем

$$\sum_{i=1}^m a_{ij} \lambda_i = c_j.$$

Поскольку x'_j — произвольная положительная компонента плана X' , вектор Λ связан с этим планом системой равенств (4.38) и, следовательно, является для него разрешающим.

Доказанное утверждение дает основание связывать понятие разрешающего вектора с полным множеством оптимальных планов данной задачи и говорить о разрешающих векторах (разрешающих множителях) задачи линейного программирования.

Заметим, что отыскание разрешающего вектора задачи V сводится к решению сопряженной с ней задачи VI.

Для доказательства этого утверждения достаточно установить совпадение множества разрешающих векторов задачи V и множества оптимальных планов задачи VI.

1. Пусть $\Lambda = (\lambda_1, \dots, \lambda_m)$ — произвольный разрешающий вектор задачи V. При доказательстве теоремы о разрешающих векторах было показано, что Λ — оптимальный план задачи VI.

2. Пусть $Y = (y_1, \dots, y_m)$ — любое решение задачи VI. В силу второй теоремы двойственности Y связан равенствами (4.38) с каждым из оптимальных планов задачи V и является, следовательно, ее разрешающим вектором.

С помощью теорем двойственности можно получить критерий оптимальности для произвольной задачи типа III (задача линейного программирования со смешанными условиями).

Назовем $\Lambda = (\lambda_1, \dots, \lambda_m)$ разрешающим вектором задачи III, если

$$\text{а) } \sum_{i=1}^m a_{ij}\lambda_i \geq c_j, \quad j=1, 2, \dots, n_1;$$

$$\text{б) } \sum_{i=1}^m a_{ij}\lambda_i = c_j, \quad j=n_1+1, \dots, n;$$

$$\text{в) } \lambda_i \geq 0, \quad i=1, 2, \dots, m_1;$$

г) для некоторого плана $X = (x_1, \dots, x_n)$ задачи III выполняются условия:

при

$$\sum_{j=1}^n a_{ij}x_j < b_i, \quad 1 \leq i \leq m_1, \quad \lambda_i = 0;$$

при

$$x_j > 0, \quad 1 \leq j \leq n_1, \quad \sum_{i=1}^m a_{ij}\lambda_i = c_j.$$

Критерий оптимальности плана задачи III. Для того чтобы план $X = (x_1, \dots, x_n)$ задачи III был оптималь-

ным, необходимо и достаточно существование разрешающего вектора $\Lambda = (\lambda_1, \dots, \lambda_m)$, связанного с этим планом условиями γ).

Справедливость сформулированного критерия устанавливается точно так же, как и в случае задачи V. Поэтому соответствующих доказательств мы приводить не будем.

Отметим, что в рассматриваемом случае можно также говорить о разрешающем векторе соответствующей задачи, поскольку разрешающий вектор некоторого оптимального плана задачи III автоматически является таковым для каждого из ее решений.

1.6. Геометрическая интерпретация задач двойственной пары. В заключение этого краткого рассмотрения теории двойственности дадим совместную геометрическую интерпретацию исходной и сопряженной задач линейного программирования. Обратимся к общей задаче линейного программирования, записанной в канонической форме (V). Сопряженная с ней задача была выписана нами ранее и обозначена через VI. Напомним, геометрический смысл задачи V в $(m+1)$ -мерном пространстве (§ 7 гл. 1). При этом читатель может ограничиться трехмерным пространством ($m=2$), в котором все сказанное ниже приобретает геометрическую наглядность.

Каждому n -мерному вектору $X = (x_1, \dots, x_n)$ поставим в соответствие $(m+1)$ -мерный вектор (точку $(m+1)$ -

мерного пространства) $z = \sum_{j=1}^n \bar{A}_j x_j$, где $\bar{A}_j = (a_{1j},$

$a_{2j}, \dots, a_{mj}, c_j)$ ($j=1, 2, \dots, n$) — расширенный вектор условий. В таком случае множеству n -мерных векторов с неотрицательными компонентами в $(m+1)$ -мерном пространстве соответствует выпуклый многогранный конус K .

Этот конус является минимальным выпуклым множеством, содержащим все лучи, выходящие из начала координат и имеющие в качестве направляющего один из векторов: $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n$ (говорят, что конус K порожден векторами $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n$). В рассматриваемом $(m+1)$ -мерном пространстве множеству векторов

X , удовлетворяющих условиям (4.28), соответствует прямая $\mathcal{L}$, параллельная оси z_{m+1} :

$$z_i = b_i, \quad i = 1, 2, \dots, m;$$

$$z_{m+1} = t, \quad -\infty < t < +\infty.$$

Планам задачи V соответствуют те и только те точки z , которые принадлежат как конусу K , так и прямой $\mathcal{L}$. Если $\mathcal{L}$ не пересекает конус K , множество планов рассматриваемой задачи пусто. Допустим, что пересечение K и $\mathcal{L}$ не пусто. Тогда оптимальному плану задачи V соответствует точка общей части K и $\mathcal{L}$ с максимальной $(m+1)$ -й координатой. Очевидно, если общая часть K и $\mathcal{L}$ является отрезком прямой, такая точка заведомо найдется. Если же общая часть конуса K и прямой $\mathcal{L}$ совпадает с полупрямой, уходящей в положительном направлении $(m+1)$ -й оси (или со всей прямой $\mathcal{L}$), линейная форма задачи V неограничена (сверху) на множестве ее планов. В силу выпуклости конуса K прямая $\mathcal{L}$ может пересекать его границу в одной или двух точках (случай касания прямой и конуса мы оставляем в стороне). Если $\mathcal{L}$ пересекает границу K в двух точках, линейная форма (4.27) при условиях (4.28), (4.29) может быть обращена как в максимум, так и в минимум, причем максимальному плану соответствует крайняя «верхняя» точка пересечения K и $\mathcal{L}$, минимальному плану — крайняя «нижняя» точка (за положительное направление на $\mathcal{L}$ принято направление оси z_{m+1}). Если же $\mathcal{L}$ пересекает границу K лишь в одной точке, линейная форма (4.37) при условиях задачи V обладает либо конечным максимумом, либо конечным минимумом в зависимости от того, «верхняя» или «нижняя» часть $\mathcal{L}$ принадлежит K .

В том же $(m+1)$ -мерном пространстве дадим геометрическую интерпретацию сопряженной задачи VI . Аналог плоскости в многомерном пространстве обычно называют гиперплоскостью. Если говорить точнее, то гиперплоскостью $(m+1)$ -мерного пространства называется множество точек $Z = (z_1, \dots, z_{m+1})$, координаты которых удовлетворяют некоторому линейному уравнению

$$\sum_{k=1}^{m+1} \lambda_k z_k = c.$$

Мы будем рассматривать гиперплоскости, содержащие начало координат. Уравнение любой такой гиперплоскости имеет вид

$$\sum_{k=1}^{m+1} \lambda_k z_k = 0. \quad (4.40)$$

Вектор $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_{m+1})$ назовем направляющим вектором гиперплоскости (4.40).

Гиперплоскость Π и ее направляющий вектор Λ связаны условием ортогональности.

Очевидно, что направляющий вектор гиперплоскости определяется с точностью до произвольного множителя. Для нашей цели удобно в каждом направляющем векторе $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_{m+1})$ считать $(m+1)$ -ю координату равной -1 ($\lambda_{m+1} = -1$). Тем самым из рассмотрения исключаются гиперплоскости, параллельные оси z_{m+1} . В этих предположениях между гиперплоскостями, содержащими начало координат, и их направляющими векторами устанавливается взаимнооднозначное соответствие.

Пусть $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m)$ — произвольный план задачи VI. В $(m+1)$ -мерном пространстве рассмотрим гиперплоскость с направляющим вектором $\Lambda = (\lambda_1, \dots, \lambda_m, -1)$. Ее уравнение имеет вид

$$\sum_{i=1}^m \lambda_i z_i - z_{m+1} = 0. \quad (4.41)$$

Проверим, что конус K лежит по одну сторону от введенной гиперплоскости. Действительно, в соответствии с условиями (4.31) координаты любого из векторов $\bar{A}_j, j=1, 2, \dots, n$, при подстановке в левую часть (4.41) дают неотрицательную величину. Следовательно, натянутый на эти векторы конус K лежит по ту же сторону гиперплоскости (4.41), что и единичный вектор $-e_{m+1} = (0, 0, \dots, 0, -1)$, соответствующий отрицательному направлению оси z_{m+1} . Поэтому можно сказать, что конус K расположен под гиперплоскостью (4.41).

Так же легко проверяется, что первые m координат направляющего вектора $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m, -1)$ любой гиперплоскости, содержащей начало координат и расположенной над K , составляют план задачи VI. Итак, в $(m+1)$ -мерном пространстве геометрическим образом множества планов задачи VI является совокупность гиперплоскостей, содержащих начало координат и расположенных над конусом K . Выясним теперь геометрический смысл оптимального плана сопряженной задачи VI. Пусть вектор $Z^* = (z_1^*, \dots, z_{m+1}^*)$ соответствует оптимальному плану исходной задачи V (верхняя крайняя точка пересечения прямой $\mathcal{L}$ и конуса K). По определению:

$$z_i^* = b_i, \quad i = 1, 2, \dots, m, \quad (*)$$

$$z_{m+1}^* = \sum_{j=1}^n c_j x_j^*, \quad \text{где } X^* = (x_1^*, \dots, x_n^*)$$

— решение задачи V. Рассмотрим гиперплоскость, проходящую через начало координат, точку $Z^* = (z_1^*, \dots, z_{m+1}^*)$, и лежащую над конусом K :

$$\sum_{i=1}^m \lambda_i^* z_i - z_{m+1} = 0.$$

Проверим, что она соответствует оптимальному плану задачи VI. В самом деле, величины $(\lambda_1^*, \lambda_2^*, \dots, \lambda_m^*)$ составляют по условию план задачи VI. Далее

$$\sum_{i=1}^m \lambda_i^* z_i^* = z_{m+1}^*.$$

Откуда, учитывая соотношения (*), получаем

$$\sum_{i=1}^m \lambda_i^* b_i = z_{m+1}^* = \sum c_i x_i^*.$$

По следствию 2 первой теоремы двойственности это равенство указывает на оптимальность плана $(\lambda_1^*, \dots, \lambda_m^*)$.

Нетрудно проверить также, что любому оптимальному плану задачи VI соответствует гиперплоскость, обладающая указанными тремя свойствами. Таким образом, классу оптимальных планов задачи VI соответствует множество гиперплоскостей, проходящих через начало координат, содержащих «верхнюю» крайнюю точку пересечения K и $\mathcal{L}$ и расположенных над конусом K .

Переходим к изложению методов линейного программирования, опирающихся на теорию двойственности.

§ 2. МЕТОД ПОСЛЕДОВАТЕЛЬНОГО УТОЧНЕНИЯ ОЦЕНОК

2.1. Предварительные замечания. Будем рассматривать общую задачу линейного программирования, записанную в форме V. Напомним ее формулировку.

Требуется определить вектор $X = (x_1, x_2, \dots, x_n)$, обращающий в максимум линейную форму

$$\sum_{j=1}^n c_j x_j \quad (4.27)$$

при наличии ограничений

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m; \quad (4.28)$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n. \quad (4.29)$$

Назовем эту задачу исходной. В предыдущем параграфе было показано, что задача, двойственная исходной задаче V, формулируется следующим образом.

Определить вектор $Y = (y_1, y_2, \dots, y_m)$, обращающий в минимум линейную форму

$$\sum_{i=1}^m b_i y_i \quad (4.30)$$

при наличии ограничений

$$\sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n. \quad (4.31)$$

Эта задача обозначалась нами цифрой VI.

Сущность метода последовательного уточнения оценок состоит в следующем. Вместо того чтобы решать непосредственно задачу V, обращаемся вначале к сопряженной с ней задаче VI, отыскиваем ее оптимальный план, а затем уже, используя двойственную связь обеих задач, получаем искомое решение исходной задачи. Таким образом, решение исходной задачи (4.27), (4.28), (4.29) сводится к определению оптимального плана сопряженной задачи (4.30), (4.31). В некоторых случаях такая редукция оказывается полезной. В предыдущем параграфе было показано, что компоненты оптимального плана сопряженной задачи VI являются оценками влияния различных уравнений системы (4.28) на величину максимума линейной формы (4.27). Этим объясняется название рассматриваемого метода, который дает возможность, исходя из приблизительных оценок условий (4.28) (исходного плана сопряженной задачи), последовательным их уточнением получить вектор окончательных оценок (оптимальный план сопряженной задачи). В ходе решения задачи методом последовательного уточнения оценок образуется ряд n -мерных векторов $X_1, X_2, \dots, X_N$, оканчивающийся оптимальным планом задачи V. При этом каждый из векторов $X_i = (x_1^{(i)}, \dots, x_n^{(i)})$, $i = 1, 2, \dots, N$ удовлетворяет системе уравнений (4.28). Что касается условий (4.29), то им удовлетворяет лишь X_N , являющийся решением задачи V.

Следует заметить, что значение линейной формы (4.27) при переходе от одного члена указанной последовательности планов к другому монотонно убывает. Таким образом, в методе последовательного уточнения оценок приближение к максимуму линейной формы (4.27) при условиях (4.28), (4.29) осуществляется не снизу (как это имело место в методе последовательного улучшения плана), а сверху.

Впервые метод уточнения оценок был описан Лемке [54] (автор назвал его двойственным методом). В работе [54] этот метод рассматривался как синтез метода последовательного улучшения плана и основных идей двойственности. С тех же позиций метод уточнения оценок излагается и в настоящем параграфе. Описание ме-

тогда проводится по следующему плану. В начале даются основы метода для невырожденного случая, затем следует описание алгоритма. В заключение мы освобождаемся от ограничительного предположения о невырожденности сопряженной задачи.

2.2. Основы метода. Введем прежде всего два понятия, относящихся к сопряженной задаче типа VI.

План сопряженной задачи (4.30), (4.31) будем называть опорным, если среди неравенств системы (4.31), обращающихся на нем в равенства, имеется m линейно независимых.

Это определение соответствует понятию опорного плана, введенному нами ранее для канонической формы задачи линейного программирования, (задача типа V). В самом деле, в обоих случаях опорными планами являются те и только те точки многогранника условий, которые расположены на пересечении k независимых гиперплоскостей из числа образующих этот многогранник, причем k совпадает с числом переменных задачи. Таким образом, множество опорных планов задачи VI совпадает с совокупностью вершин многогранника, высекаемого в m -мерном пространстве условиями (4.31). Дадим теперь применительно к рассматриваемой задаче VI определение невырожденности.

Сопряженную задачу (4.30), (4.31) назовем невырожденной, если на каждом из ее опорных планов в равенства обращаются ровно m условий системы (4.31). Как нетрудно заметить, геометрический смысл ограничений, гарантирующих невырожденность задачи VI, заключается в требовании, чтобы в каждой вершине многогранника условий этой задачи пересекалось ровно m граничных гиперплоскостей. Поэтому введенное понятие полностью соответствует определению невырожденности для задач линейного программирования в канонической форме, данному нами в § 7 гл. 3. В рассмотрении этого и следующего пунктов задача (4.30), (4.31) предполагается невырожденной. Случай вырожденности будет разобран особо. Допустим, что нам известен некоторый спорный план $Y = (y_1, y_2, \dots, y_n)$ задачи VI. Если план Y оптимальный, изучаемый метод дает возможность обнаружить это. В противном случае метод позволяет либо убедиться в неразрешимости задачи V, либо перейти

к новому опорному плану задачи VI, связанному с уменьшенным значением линейной формы (4.30). Совокупность соответствующих операций назовем итерацией или шагом метода последовательного уточнения оценок. Поскольку число опорных планов задачи VI конечно, оптимальный план этой задачи (если он существует) будет обнаружен за ограниченное число итераций. При этом автоматически будет получен оптимальный план исходной задачи V. По существу метод уточнения оценок заключается в применении метода улучшения плана к сопряженной задаче. Поэтому его можно было бы получить с помощью небольших изменений алгоритма улучшения плана, связанных с особым видом задачи (4.30), (4.31). Однако мы считаем полезным дать для метода оценок независимое изложение.

Систему линейно независимых векторов

$$A_{jk} = (a_{1jk}, a_{2jk}, \dots, a_{mjk}), \quad k = 1, 2, \dots, m$$

назовем базисом опорного плана $Y = (y_1, \dots, y_m)$ сопряженной задачи VI, если

$$\sum_{i=1}^m a_{ij_k} y_i = c_{j_k}, \quad k = 1, 2, \dots, m.$$

Из предположения о невырожденности задачи VI получаем

$$\sum_{i=1}^m a_{ij} y_i > c_j \quad \text{для } j \neq j_k, \quad k = 1, 2, \dots, m.$$

Для упрощения индексации положим, что базис исходного опорного плана $Y = (y_1, \dots, y_m)$ состоит из векторов $A_1, A_2, \dots, A_m$. Это допущение, конечно, несколько не снизит общности наших рассуждений.

Критерий оптимальности плана. Если среди величин $x_1, x_2, \dots, x_m$, определяемых из системы уравнений

$$\sum_{j=1}^m a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m, \quad (4.42)$$

нет отрицательных, то $Y = (y_1, y_2, \dots, y_m)$ является оптимальным планом задачи VI, а $X = (x_1, \dots, x_m, 0, \dots, 0)$ — решением исходной задачи V.

Сформулированное утверждение является частным случаем критерия оптимальности для задач со смешанными условиями ($n_1 = 0, m_1 = m$). Поэтому наличие неотрицательного решения системы (4.42) не только достаточно, но и необходимо для оптимальности плана Y . Здесь мы не будем опираться на общий критерий оптимальности и приведем независимое доказательство.

Доказательство. Пусть решение системы (4.42) составлено из неотрицательных величин $x_1, x_2, \dots, x_m$. Пользуясь системой равенств (4.42), получаем

$$\sum_{i=1}^m b_i y_i = \sum_{i=1}^m \left(\sum_{j=1}^m a_{ij} x_j \right) y_i = \sum_{j=1}^m x_j \left(\sum_{i=1}^m a_{ij} y_i \right).$$

Но по условию

$$\sum_{i=1}^m a_{ij} y_i = c_j, \quad j = 1, 2, \dots, m,$$

Следовательно,

$$\sum_{i=1}^m b_i y_i = \sum_{j=1}^m c_j x_j.$$

Полагая теперь $x_j = 0$ для $j = m + 1, \dots, n$, приходим к равенству

$$\sum_{i=1}^m b_i y_i = \sum_{j=1}^n c_j x_j. \quad (4.43)$$

По условию $Y = (y_1, y_2, \dots, y_m)$ — план задачи VI. Из неотрицательности $x_j, j = 1, 2, \dots, n$ и системы равенств (4.42) вытекает, что $X = (x_1, x_2, \dots, x_m, 0, 0, \dots, 0)$ — план исходной задачи V. Но согласно следствию 2 первой теоремы двойственности планы взаимосопряженных задач, связанные соотношением (4.43), обязаны быть оптимальными. Тем самым справедливость критерия установлена.

Итак, если все компоненты решения системы (4.42), связанной с планом задачи VI, оказываются неотрицательными, получаем оптимальный опорный план исходной задачи $X = (x_1, x_2, \dots, x_m, 0, \dots, 0)$ с базисом $A_1, A_2, \dots, A_m$.

Предположим теперь, что среди величин $x_1, \dots, x_m$ имеются отрицательные.

Обозначим через e_{st} , $t = 1, 2, \dots, m$ совокупность величин, удовлетворяющих системе уравнений

$$\sum_{i=1}^m a_{ij} e_{si} = \begin{cases} 0, & j \neq s \\ 1, & j = s \end{cases} \quad (1 \leq j \leq m). \quad (4.44)$$

Придавая индексу s значения от 1 до m , получаем квадратную матрицу $\|e_{st}\|$ порядка m .

Из элементов матрицы $\|e_{st}\|$ можно составить коэффициенты разложения любого из m -мерных единичных векторов e_t , $t = 1, 2, \dots, m$, по базису $A_1, A_2, \dots, A_m$. Действительно, пусть

$$e_t = \sum_{j=1}^m \lambda_j^{(t)} A_j.$$

Перепишем это соотношение в координатах

$$\sum_{j=1}^m \lambda_j^{(t)} a_{ij} = \begin{cases} 0, & i \neq t, \\ 1 & i = t. \end{cases} \quad (4.45)$$

Умножая i -е равенство системы (4.45) на e_{si} и складывая результаты, имеем

$$\sum_{i=1}^m e_{si} \left(\sum_{j=1}^m \lambda_j^{(t)} a_{ij} \right) = e_{st}.$$

С другой стороны, меняя порядок суммирования и учитывая систему равенств (4.44), получаем

$$\sum_{i=1}^m e_{si} \left(\sum_{j=1}^m \lambda_j^{(t)} a_{ij} \right) = \sum_{j=1}^m \lambda_j^{(t)} \sum_{i=1}^m a_{ij} e_{si} = \lambda_s^{(t)}.$$

Итак, $\lambda_s^{(t)} = e_{st}$, $s = 1, 2, \dots, m$, т. е. коэффициенты разложения вектора e_t по базису $A_1, \dots, A_m$ расположены в t -м столбце матрицы

$$\|e_{st}\| = \begin{vmatrix} e_{11} & e_{12} & \dots & e_{1t} & \dots & e_{1m} \\ e_{21} & e_{22} & \dots & e_{2t} & \dots & e_{2m} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ e_{s1} & e_{s2} & \dots & e_{st} & \dots & e_{sm} \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ e_{m1} & e_{m2} & \dots & e_{mt} & \dots & e_{mm} \end{vmatrix}^*.$$

Элементы матрицы $\|e_{st}\|$ можно использовать для разложения произвольного вектора

$$W = (\varpi_1, \varpi_2, \dots, \varpi_m)$$

по векторам нашего базиса. Для этого следует заметить, что

$$W = \sum_{j=1}^m \varpi_j e_j,$$

и воспользоваться полученным разложением единичных векторов $e_1, e_2, \dots, e_m$

$$\begin{aligned} W &= \sum_{j=1}^m \varpi_j e_j = \sum_{j=1}^m \varpi_j \left(\sum_{i=1}^m e_{ij} A_i \right) = \\ &= \sum_{i=1}^m \left(\sum_{j=1}^m \varpi_j e_{ij} \right) A_i = \sum_{i=1}^m \bar{\varpi}_i A_i. \end{aligned}$$

Таким образом,

$$W = \sum_{i=1}^m \bar{\varpi}_i A_i, \quad ,$$

где

$$\bar{\varpi}_i = \sum_{j=1}^m e_{ij} \varpi_j. \quad (4.46)$$

^{*)} Читатель, знакомый с понятием обратной матрицы, без труда заметит, что $\|e_{st}\|_{m,m} = \|a_{ij}\|_{m,m}^{-1}$. Здесь через $\|a_{ij}\|_{m,m}^{-1}$ обозначена матрица, обратная $\|a_{ij}\|_{m,m}$.

Имея матрицу $\|e_{st}\|$, нетрудно также решить систему уравнений

$$\sum_{i=1}^m a_{ij}v_i = d_j, \quad j = 1, 2, \dots, m,$$

при любых правых частях

$$d_1, d_2, \dots, d_m.$$

Проверим, что решение этой системы определяется следующим образом:

$$v_i = \sum_{s=1}^m e_{si}d_s, \quad i = 1, 2, \dots, m. \quad (4.46')$$

Действительно, используя равенства (4.44), получаем:

$$\begin{aligned} \sum_{i=1}^m a_{ij}v_i &= \sum_{i=1}^m a_{ij} \sum_{s=1}^m e_{si}d_s = \sum_{s=1}^m d_s \sum_{i=1}^m a_{ij}e_{si} = \\ &= d_1 \cdot 0 + d_2 \cdot 0 + \dots + d_{j-1} \cdot 0 + \\ &+ d_j + d_{j+1} \cdot 0 + \dots + d_m \cdot 0 = d_j. \end{aligned}$$

После этих предварительных замечаний приступим к построению нового опорного плана задачи VI.

Пусть $x_k = \min_{1 \leq j \leq m} x_j$ (по предположению $x_k < 0$).

Введем в рассмотрение семейство векторов

$$Y(\theta) = Y + \theta e^{(k)},$$

где

$$e^{(k)} = (e_{k1}, e_{k2}, \dots, e_{km})$$

(компоненты вектора $e^{(k)}$ расположены в k -й строке матрицы $\|e_{st}\|$), $\theta > 0$.

В зависимости от знаков величин

$$z_{kj} = \sum_{i=1}^m a_{ij} e_{ki}, \quad j = m+1, m+2, \dots, n \quad (4.47)$$

будем различать случаи 1°, 2°:

1°. Среди z_{kj} имеются отрицательные.

2°. $z_{kj} \geq 0, j = m+1, m+2, \dots, n$.

Разберем каждый из этих случаев отдельно.

1°. Покажем, что при некотором значении θ ($\theta = \theta_0$) $Y(\theta)$ является опорным планом задачи VI, уменьшающим по сравнению с Y значение линейной формы (4.30).

Подставляя компоненты вектора $Y(\theta) = (y_1(\theta), \dots, y_m(\theta))$ в условия сопряженной задачи VI, получаем

$$\sum_{i=1}^m a_{ij} y_i(\theta) = \sum_{i=1}^m a_{ij} y_i + \theta \cdot \sum_{i=1}^m a_{ij} e_{ki}; \quad j = 1, 2, \dots, n.$$

В соответствии с определением величин e_{st} (4.44)

$$\sum_{i=1}^m a_{ij} e_{ki} = \begin{cases} 0, & j \neq k \quad (1 \leq j \leq m), \\ 1, & j = k. \end{cases}$$

Поэтому

$$\sum_{i=1}^m a_{ij} y_i(\theta) = \sum_{i=1}^m a_{ij} y_i, \quad (4.48)$$

$$j = 1, 2, \dots, k-1, k+1, \dots, m;$$

$$\sum_{i=1}^m a_{ik} y_i(\theta) = \sum_{i=1}^m a_{ik} y_i + \theta. \quad (4.49)$$

Что касается $j = m+1, m+2, \dots, n$, то по определению величин z_{kj} (4.47)

$$\sum_{i=1}^m a_{ij} y_i(\theta) = \sum_{i=1}^m a_{ij} y_i + \theta z_{kj}, \quad j = m+1, \dots, n. \quad (4.50)$$

Положим $\Delta_j = \sum_{i=1}^m a_{ij}y_i - c_j$ и через θ_0 обозначим $\theta_0 = \min_{z_{kj} < 0} \left(-\frac{\Delta_j}{z_{kj}} \right)$, где минимум берется по тем j , для которых $z_{kj} < 0$ (множество таких j по предположению не пусто). Как уже отмечалось, при $j = m+1, m+2, \dots, n$ $\Delta_j > 0$ (сопряженная задача предполагается невырожденной). Следовательно, $\theta_0 > 0$.

Определим индекс l условием

$$\theta_0 = \min_{z_{kj} < 0} \left(-\frac{\Delta_j}{z_{kj}} \right) = -\frac{\Delta_l}{z_{kl}}. \quad (4.51)$$

Из соотношений (4.48), (4.49), (4.50) и определения величины θ_0 получаем, что вектор $Y(\theta_0)$ является планом задачи VI, обращающим в равенства условия системы (4.31) с номерами $1, 2, \dots, k-1, k+1, \dots, m, l$.

Для доказательства опорности введенного плана убедимся в линейной независимости векторов $A_1, A_2, \dots, A_{k-1}, A_{k+1}, \dots, A_m, A_l$.

Пусть

$$\sum_{\substack{j=1 \\ j \neq k}}^m d_j A_j + d_l A_l = 0. \quad (4.52)$$

Раскладывая вектор A_l по базису $A_1, A_2, \dots, A_m$

$$A_l = \sum_{j=1}^m r_j A_j$$

и подставляя полученное выражение в (4.52), имеем

$$\sum_{\substack{j=1 \\ j \neq k}}^m (d_j + r_j d_l) A_j + r_k d_l A_k = 0.$$

Поскольку векторы $A_1, A_2, \dots, A_k, \dots, A_m$ линейно независимы,

$$d_j + r_j d_l = 0, \quad j = 1, 2, \dots, k-1, k+1, \dots, m; \quad (4.53)$$

$$r_k d_l = 0. \quad (4.54)$$

Обратившись к соотношению (4.46), обнаруживаем, что согласно определению величин z_{kj}

$$r_k = z_{kl}.$$

Учитывая далее отрицательность z_{kl} и используя последовательно равенства (4.54) и (4.53), получаем

$$d_1 = d_2 = \dots = d_{k-1} = d_k = \dots = d_m = d_l = 0.$$

Итак, равенство (4.52) возможно лишь при нулевых коэффициентах d_j , что означает линейную независимость векторов $A_1, A_2, \dots, A_{k-1}, A_{k+1}, \dots, A_m, A_l$, составляющих базис нового плана $Y(\theta_0)$.

Заметим, что в предположении о невырожденности задачи VI индекс l определяется однозначно условием (4.51). В самом деле, если бы это было не так, опорный план $Y(\theta_0)$ обратил бы в равенства более чем m неравенств системы (4.31).

Оценим теперь значение линейной формы (4.30) на вновь построенном плане $Y(\theta_0)$.

Прежде всего заметим, что согласно соотношению (4.46) коэффициент при A_k в разложении вектора ограничений B по базису $A_1, A_2, \dots, A_k, \dots, A_m$ совпадает с $\sum_{i=1}^m b_i e_{ki}$, т. е.

$$x_k = \sum_{i=1}^m b_i e_{ki}.$$

Учитывая затем положительность θ_0 и отрицательность x_k , получаем

$$\begin{aligned} \sum_{i=1}^m b_i y_i(\theta_0) &= \sum_{i=1}^m b_i y_i + \theta_0 \sum_{i=1}^m b_i e_{ki} = \\ &= \sum_{i=1}^m b_i y_i + \theta_0 x_k < \sum_{i=1}^m b_i y_i. \end{aligned}$$

Таким образом, переход к новому опорному плану $Y(\theta_0)$ уменьшает линейную форму сопряженной задачи VI.

Переходим к анализу случая 2°.

2°. Если $z_{kj} \geq 0$ для $j = m+1, \dots, n$, то в соответствии с (4.48), (4.49), (4.50) $Y(\theta)$ является планом задачи VI при любом $\theta \geq 0$.

Учитывая теперь равенство

$$\sum_{i=1}^m b_i y_i(\theta) = \sum_{i=1}^m b_i y_i + \theta x_k$$

и отрицательность x_k , приходим к выводу о неограниченности снизу линейной формы задачи VI на множестве ее планов. Согласно следствию 1 первой теоремы двойственности это возможно лишь при отсутствии планов у исходной задачи V.

Таким образом, наличие случая 2° указывает на неразрешимость исследуемой задачи.

Повторим кратко последовательность действий, составляющих отдельную итерацию метода. Отправляясь от опорного плана $Y = (y_1, \dots, y_m)$ с базисом $A_1, A_2, \dots, A_m$, определяем из системы уравнений (4.42) величины $x_1, x_2, \dots, x_m$. Если все x_i неотрицательны, вектор $X = (x_1, \dots, x_m, 0, \dots, 0)$ является оптимальным планом исходной задачи V. В противном случае выбираем наименьшее из x_i , скажем x_k , и определяем из системы уравнений (4.44) (полагая $s = k$) систему величин $e_{ki}, i = 1, 2, \dots, m$.

После этого, пользуясь формулами (4.47), вычисляем $z_{kj}, j = m + 1, m + 2, \dots, n$.

Если среди z_{kj} обнаруживаются отрицательные величины (случай 1°), переходим к новому опорному плану

$$Y' = Y(\theta_0) = Y + \theta_0 e^{(k)},$$

где θ_0 определяется соотношением (4.51), а вектор

$$e^{(k)} = (e_{k1}, e_{k2}, \dots, e_{km}).$$

При этом соотношение (4.51) определяет некоторый индекс l . Базис нового плана Y' образуется из старого базиса $A_1, \dots, A_k, \dots, A_m$ путем замены вектора A_k вектором A_l .

Новый опорный план Y' приводит (в невырожденном случае) к уменьшению линейной формы (4.30).

Если же $z_{kj} \geq 0, j = m + 1, m + 2, \dots, n$ (случай 2°), множество планов исходной задачи пусто и процесс решения следует прекратить.

При использовании метода последовательного уточнения оценок мы не можем вернуться к уже пройденному один раз базису.

Действительно, предположение противного означало бы совпадение опорных планов задачи VI, полученных на различных итерациях метода, что противоречит монотонному убыванию линейной формы (4.30). Отсюда количество итераций, необходимых для решения задачи V (или обнаружения ее неразрешимости), заведомо ограничено общим числом базисов сопряженной задачи VI, не превышающим C_n^m .

2.3. Геометрическая интерпретация метода. В конце предыдущего параграфа была рассмотрена геометрическая интерпретация в $(m+1)$ -мерном пространстве пары взаимосопряженных задач линейного программирования (V, VI). Поясним теперь, оставаясь в том же пространстве, геометрический смысл метода последовательного уточнения оценок. В этом пространстве каждому опорному плану сопряженной задачи соответствует гиперплоскость Π , проходящая через начало координат, содержащая те из ребер конуса K , которые (вернее, совокупности первых m координат которых) составляют базис рассматриваемого плана, и расположенная вместе с тем над K . Таким образом, гиперплоскость, соответствующая опорному плану задачи VI, обязана содержать некоторую m -мерную грань конуса K . Если вектор, проведенный в точку, соответствующую оптимальному плану исходной задачи V, принадлежит указанной грани, гиперплоскость Π определяет оптимальный план сопряженной задачи VI. В противном случае точка пересечения прямой $\mathcal{L}$ ($z_i = b_i, i = 1, 2, \dots, m, z_{m+1} = t$) с гиперплоскостью Π расположена над конусом K (в смысле оси z_{m+1}).

Допустим, что гиперплоскость Π соответствует некоторому опорному плану задачи VI с базисом $A_1, A_2, \dots, A_k, \dots, A_m$. Метод уточнения оценок позволяет выделить из системы векторов $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_k, \dots, \bar{A}_m$ *) такой вектор $\bar{A}_k$, что поворот гиперплоскости Π вокруг оставшихся векторов $\bar{A}_1, \dots, \bar{A}_{k-1}, \bar{A}_{k+1}, \dots, \bar{A}_m$ приводит к уменьшению $(m+1)$ -й координаты точки пересечения $\mathcal{L}$ и Π .

*) $\bar{A}_i = (a_{j1}, a_{j2}, \dots, a_{jm}, c_j)$.

Поворот осуществляется до тех пор, пока гиперплоскость не захватит один из векторов $\bar{A}_{m+1}, \bar{A}_{m+2}, \dots, \bar{A}_r$ (дальнейший поворот привел бы к гиперплоскости, не соответствующей плану задачи VI). Новая гиперплоскость Π_1 соответствует опорному плану сопряженной задачи, связанному с уменьшенным значением ее линейной формы [($m+1$)-я координата точки пересечения $\mathcal{L}$ и Π_1]. Процесс последовательных поворотов проводится до получения такой гиперплоскости Π^* , точка пересечения которой с прямой $\mathcal{L}$ расположена на границе конуса K . Эта гиперплоскость отвечает оптимальному плану сопряженной задачи, а точка пересечения Π^* и $\mathcal{L}$ дает решение исходной задачи V.

Таким образом, в методе последовательного уточнения оценок приближение к оптимальному плану задачи V (крайней верхней точки пересечения K и $\mathcal{L}$) осуществляется извне конуса K (этим приближениям соответствуют точки пересечения прямой $\mathcal{L}$ с промежуточными гиперплоскостями, отвечающими планам задачи VI).

2.4. Алгоритм метода. Описанная выше совокупность действий, позволяющая получить решение исходной задачи последовательным переходом от одного опорного плана сопряженной задачи к другому, может быть представлена в виде компактной схемы, составляющей алгоритм метода уточнения оценок.

Условия исходной и сопряженной задач записываются в первых $m+1$ строках вспомогательной табл. II. При этом первая строка таблицы занимается коэффициентами линейной формы задачи V, ($n+1$)-й столбец — координатами вектора ограничений $B = (b_1, b_2, \dots, b_m)$.

На пересечении ($i+1$)-й строки и j -го столбца табл. II помещается элемент a_{ij} . В процессе решения задачи вспомогательная таблица продолжается вниз и вправо.

Допустим, что нам известен некоторый опорный план сопряженной задачи $Y = (y_1, y_2, \dots, y_m)$ с базисом $A_1, A_2, \dots, A_m$ (вопрос о построении исходного плана рассматривается ниже). Прежде всего определим, решающая системы уравнений (4.44) для $s = 1, 2, \dots, m$, величины e_{st} , $t = 1, 2, \dots, m$.

Таблица II

A	A ₁	A ₂	...	A _{k-1}	A _k	A _{k+1}	...	A _m	A _{m+1}	...	A _{l'}	...	A _l	...	A _n	B	A _k	A _{k'}	...
c	c ₁	c ₂	...	c _{k-1}	c _k	c _{k+1}	...	c _m	c _{m+1}	...	c _{l'}	...	c _l	...	c _n				
	a ₁₁	a ₁₂	...	a _{1,k-1}	a _{1k}	a _{1,k+1}	...	a _{1m}	a _{1,m+1}	...	a _{1l'}	...	a _{1l}	...	a _{1n}	b ₁	e _{k1}	e' _{k'1}	...
	a ₂₁	a ₂₂	...	a _{2,k-1}	a _{2k}	a _{2,k+1}	...	a _{2m}	a _{2,m+1}	...	a _{2l'}	...	a _{2l}	...	a _{2n}	b ₂	e _{k2}	e' _{k'2}	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	a _{i1}	a _{i2}	...	a _{i,k-1}	a _{ik}	a _{i,k+1}	...	a _{im}	a _{i,m+1}	...	a _{il'}	...	a _{il}	...	a _{in}	b _i	e _{ki}	e' _{k'i}	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
	a _{m-1,1}	a _{m-1,2}	...	a _{m-1,k-1}	a _{m-1,k}	a _{m-1,k+1}	...	a _{m-1,m}	a _{m-1,m+1}	...	a _{m-1,l'}	...	a _{m-1,l}	...	a _{m-1,n}	b _{m-1}	e _{k,m-1}	e' _{k',m-1}	...
	a _{m1}	a _{m2}	...	a _{m,k-1}	a _{m,k}	a _{m,k+1}	...	a _{mm}	a _{m,m+1}	...	a _{ml'}	...	a _{ml}	...	a _{mn}	b _m	e _{km}	e' _{k'm}	...

Продолжение табл. II

A	A ₁	A ₂	...	A _{k-1}	A _k	A _{k+1}	...	A _m	A _{m+1}	...	A _l	...	A _n	B	A _k	A _k '	...
c	c ₁	c ₂	...	c _{k-1}	c _k	c _{k+1}	...	c _m	c _{m+1}	...	c _l	...	c _n				
Δ			...				...		Δ _{m+1}	...	Δ _l	...	Δ _n				
z	z _{k1}	z _{k2}	...	z _{k, k-1}	z _{k, k}	z _{k, k+1}	...	z _{km}	z _{k, m+1}	...	z _{kl}	...	z _{kn}				
$\frac{\Delta}{z}$			...				...		$-\frac{\Delta_{m+1}}{z_{k, m+1}}$	...	$\frac{\Delta_l}{z_{kl}}$	...	$-\frac{\Delta_n}{z_{kn}}$				
Δ'			...		Δ' _k		...		Δ' _{m+1}	...	Δ' _l	...	Δ' _n				
z'	z' _{k'1}	z' _{k'2}	...	z' _{k', k-1}	z' _{k'k}	z' _{k', k+1}	...	z' _{k'm}	z' _{k', m+1}	...	z' _{k'l}	...	z' _{k'n}				
$\frac{\Delta'}{z'}$			...		$\frac{\Delta'_k}{z'_{k'k}} - \frac{\Delta'_k}{z'_{k'k}}$		...		$-\frac{\Delta'_{m+1}}{z'_{k', m+1}}$	...	$\frac{\Delta'_l}{z'_{k'l}}$	...	$-\frac{\Delta'_n}{z'_{k'n}}$				
Δ''			...				...			...		...					
z''			...				...			...		...					
$\frac{\Delta''}{z''}$			...				...			...		...					
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...

Как было показано выше, того же результата можно добиться разложением единичных векторов $e_1, e_2, \dots, e_m$ по базису $A_1, A_2, \dots, A_m$ *). Элементами матрицы $\|e_{st}\|$ заполняется основная часть табл. Ia (строки $A_1, A_2, \dots, A_m$; столбцы $e_1, e_2, \dots, e_m$). В 1-й столбец табл. Ia (столбец e_0) помещаются коэффициенты разложения вектора ограничений B по базису исходного плана Y .

В соответствии с формулами (4.46)

$$x_i = \sum_{t=1}^m e_{it} b_t. \tag{4.55}$$

Для однообразия записи x_i обозначается через e_{i0}

Т а б л и ц а I a

$B_1 \backslash e$	e_0	e_1	e_2	$\dots$	e_l	$\dots$	e_m	A_l
A_1	e_{10}	e_{11}	e_{12}	$\dots$	e_{1l}	$\dots$	e_{1m}	z_{1l}
A_2	e_{20}	e_{21}	e_{22}	$\dots$	e_{2l}	$\dots$	e_{2m}	z_{2l}
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
A_k	e_{k0}	e_{k1}	e_{k2}	$\dots$	e_{kl}	$\dots$	e_{km}	z_{kl}
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$\vdots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
A_m	e_{m0}	e_{m1}	e_{m2}	$\dots$	e_{ml}	$\dots$	e_{mm}	z_{ml}
	$e_{m+1,0}$	$e_{m+1,1}$	$e_{m+1,2}$	$\dots$	$e_{m+1,l}$	$\dots$	$e_{m+1,m}$	Δ_l

В $(m+1)$ -ю строку табл. Ia помещаем значение линейной формы задачи VI на плане Y

$$e_{m+1,0} = \sum_{i=1}^m b_i y_i,$$

*) Матрица $\|e_{st}\|$ является, очевидно, результатом обращения матрицы $\|A_1, A_2, \dots, A_m\|$, составленной из векторов нашего базиса. Поэтому для ее вычисления можно применить любой из известных численных методов линейной алгебры (см., например, [17]).

а также компоненты этого плана $y_i = e_{m+1,i}$, $i = 1, 2, \dots, m$. Заметим, что в соответствии с формулами (4.46) величины $e_{m+1,i}$ можно получить из элементов e_{st} , $1 \leq s, t \leq m$:

$$y_i = e_{m+1,i} = \sum_{s=1}^m c_s e_{si}, \quad i = 1, 2, \dots, m. \quad (4.56)$$

Если элементы $e_{10}, e_{20}, \dots, e_{m0}$, расположенные в столбце e_0 , неотрицательны, оптимальный план исходной задачи V имеет вид $X = (e_{10}, e_{20}, \dots, e_{m0}, 0, 0, \dots, 0)$. Если же среди чисел e_{i0} , $i = 1, 2, \dots, m$ имеются отрицательные, выбираем из них минимальное, скажем e_{k0} , и выделяем ($\leftarrow$) k -ю строку (строку A_k) табл. Ia (вектор A_k подлежит выводу из базиса). Затем приступаем к продолжению вспомогательной табл. II. В строку Δ этой таблицы помещаем невязки плана Y

$$\Delta_j = \sum_{i=1}^m a_{ij} y_i - c_j. \quad (4.57)$$

В следующей строке z выписываем величины

$$z_{kj} = \sum_{i=1}^m a_{ij} e_{ki}. \quad (4.58)$$

Элемент z_{kj} образуется перемножением соответствующих элементов строки A_k табл. Ia и j -го столбца табл. II с последующим сложением результатов. Поэтому, прежде чем заполнять строку z , имеет смысл выписать элементы строки A_k табл. Ia справа от вспомогательной табл. II (в столбец A_k). Заметим, что при $j = 1, 2, \dots, k-1, k+1, \dots, m$

$$z_{kj} = 0, \quad z_{kk} = 1.$$

Тем не менее величины z_{kj} , соответствующие базису плана Y , полезно для контроля вычислять также по формулам (4.58).

В строку $-\frac{\Delta}{z}$ выписываются отношения соответствующих элементов двух предшествующих строк, взятых со знаком минус $\left(-\frac{\Delta_j}{z_{kj}}\right)$, причем заполняются лишь те позиции строки, для которых $z_{kj} < 0$ (если

окажется, что отрицательных z_{kj} нет, задача V неразрешима). Затем из условия

$$-\frac{\Delta_l}{z_{kl}} = \min_{z_{kj} < 0} \left(-\frac{\Delta_j}{z_{kj}} \right) \quad (4.59)$$

определяется индекс l , указывающий, какой из векторов условий следует ввести в новый базис. В невырожденном случае соотношение (4.59) определяет l однозначно (способы однозначного выделения l в общем случае будут указаны ниже). Теперь необходимо возвратиться к основной табл. Ia. В последний столбец этой таблицы (столбец A_l) помещаем коэффициенты разложения выбранного вектора A_l по базису $A_1, A_2, \dots, A_m$:

$$z_{1l}, z_{2l}, \dots, z_{ml}.$$

В соответствии с формулами (4.46)

$$z_{il} = \sum_{t=1}^m e_{it} a_{tl}, \quad i = 1, 2, \dots, m. \quad (4.60)$$

Таким образом, i -й элемент столбца A_l образуется умножением i -й строки табл. Ia на столбец координат вводимого вектора A_l (столбец A_l табл. II).

Наконец, в правый нижний угол табл. Ia помещаем Δ_l (из строки Δ табл. II).

После завершения построения первой основной табл. Ia переходим к образованию аналогичной таблицы (табл. Ib) для нового базиса $A_1, A_2, \dots, A_{k-1}, A_l, A_{k+1}, \dots, A_m$ и заполнению последующих трех строк вспомогательной табл. II.

Начнем с табл. Ib. Элементы этой таблицы (исключая содержание последнего столбца) обозначим через $e'_{st}, s = 1, 2, \dots, m+1; t = 0, 1, 2, \dots, m$ (величина e'_{st} имеет такое же отношение к новому базису $A_1, A_2, \dots, A_l, \dots, A_m$, какое e_{st} имела к старому базису $A_1, A_2, \dots, A_k, \dots, A_m$). Таким образом, первые m строк и $m+1$ столбцов табл. Ib заполняются коэффициентами разложения векторов $B, e_1, e_2, \dots, e_m$ по новому базису $A_1, A_2, \dots, A_l, \dots, A_m$. Поскольку разложения этих векторов и вектора A_l по старому базису

$A_1, A_2, \dots, A_k, \dots, A_m$ нам известны (они составляют содержание первых m строк табл. 1а), для определения e'_{st} , $1 \leq s \leq m$, $0 \leq t \leq m$ можно воспользоваться выведенными в § 5 гл. 3 рекуррентными формулами (3.35)

$$e'_{lt} = \frac{1}{z_{kl}} e_{kt}, \quad t = 0, 1, \dots, m,$$

$$e'_{st} = e_{st} - \frac{z_{sl}}{z_{kl}} e_{kt}, \quad t = 0, 1, \dots, m, \quad (4.61)$$

$$s = 1, 2, \dots, k-1, k+1, \dots, m.$$

Итак, строка A_l табл. 1б образуется из строки A_k табл. 1а (вектор A_k из базиса выводится!) делением ее элементов на лежащий в той же строке элемент $(m+1)$ -го столбца z_{kl} . Строка A_s , $s \neq l$ новой таблицы получается из соответствующей строки старой таблицы с помощью вычитания из нее строки A_l , каждый элемент которой предварительно умножается на z_{sl} (z_{sl} расположен в s -й строке $(m+1)$ -го столбца табл. 1а). Что касается $(m+1)$ -й строки табл. 1б, то ее элементы можно определить по рекуррентным формулам, подобным (3.37).

$$e'_{m+1,t} = e_{m+1,t} - \frac{\Delta_l}{z_{kl}} e_{kt}, \quad t = 0, 1, \dots, m \quad (4.62)$$

(вывод этих формул ничем не отличается от приведенного в § 5 гл. 3).

Таким образом, правила построения последней строки табл. 1б полностью совпадают с правилами заполнения остальных строк этой таблицы (ведь величина Δ_l помещена в $(m+1)$ -й строке последнего столбца старой таблицы).

Заметим, что для контроля имеет смысл подсчитать элементы последней строки также и по формулам (4.56). Если среди элементов столбца e_0 табл. 1б (без учета $e_{m+1,0}$) нет отрицательных, оптимальным планом задачи V является вектор

$$X^* = (e'_{1,0}; \dots, e'_{k-1,0}; 0, e'_{k+1,0}; \dots, e'_{m,0}; 0, \dots, 0, e'_{l,0}; 0, \dots, 0).$$

В этом случае последняя строка таблицы оказывается заполненной величиной максимума линейной формы

(4.27), совпадающей с минимумом линейной формы (4.30), и компонентами решения сопряженной задачи VI

$$e'_{m+1,0}, e'_{m+1,1}, \dots, e'_{m+1,m}.$$

Если же среди элементов e'_{s0} , $s=1, 2, \dots, m$, имеются отрицательные, то из них выбирается минимальный, скажем $e'_{k'0}$, и вся описанная последовательность операций повторяется исходя из новой основной табл. Iб.

Таблица Iб

$B_s \backslash e$	e_0	e_1	e_2	$\dots$	e_l	$\dots$	e_m	$A_{e'}$
A_1	e'_{10}	e'_{11}	e'_{12}	$\dots$	e'_{1l}	$\dots$	e'_{1m}	$z_{1l'}$
A_2	e'_{20}	e'_{21}	e'_{22}	$\dots$	e'_{2l}	$\dots$	e'_{2m}	$z_{2l'}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
$\leftarrow A_{k'}$	$e'_{k'0}$	$e'_{k'1}$	$e'_{k'2}$	$\dots$	$e'_{k'l}$	$\dots$	$e'_{k'm}$	$z_{k'l'}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
A_m	e'_{m0}	e'_{m1}	e'_{m2}	$\dots$	e'_{mi}	$\dots$	e'_{mm}	$z_{ml'}$
	$e'_{m+1,0}$	$e'_{m+1,1}$	$e'_{m+1,2}$	$\dots$	$e'_{m+1,i}$	$\dots$	$e'_{m+1,m}$	$\Delta_{l'}$

Следует лишь отметить, что строка Δ' образуется из строки Δ прибавлением к ней строки z , каждый элемент которой предварительно умножается на $-\frac{\Delta_l}{z_{kl}}$ (l -й элемент строки $-\frac{\Delta}{z}$):

$$\Delta'_j = \Delta_j + \theta_0 z_{kj}, \quad j=1, 2, \dots, n; \quad \theta_0 = -\frac{\Delta_l}{z_{kl}}.$$

Процесс образования основных табл. Ia и Ib, ... и продолжения вспомогательной табл. II приводит, в конце концов, к построению оптимальных планов обеих взаимосопряженных задач.

2.5. Способы отыскания первого приближения. До сих пор мы предполагали известным некоторый опорный план

сопряженной задачи VI. Вообще говоря, отыскать план задачи линейного программирования ничуть не легче, чем, отправляясь от известного плана, получить ее решение. Однако во многих практических задачах построение первоначального плана осуществляется довольно просто. Если, например, одна и та же компонента всех векторов условий задачи V положительна, отыскание плана сопряженной задачи VI не составляет труда. В самом деле, пусть подобной компонентой является k -я. В таком случае при достаточно большой величине y_k

$$\left(y_k \geq \max_{1 \leq j \leq n} \frac{c_j}{a_{kj}} \right) \quad \text{вектор} \quad Y = \overbrace{(0, 0, \dots, 0, y_k, 0, \dots, 0)}^k$$

удовлетворяет, очевидно, системе неравенств (4.31) и, следовательно, является планом задачи VI. К сожалению, определенный таким образом план обычно не обладает свойством опорности. Поэтому для того, чтобы применить метод последовательного уточнения оценок, необходимо перейти от него к одному из опорных планов сопряженной задачи. Необходимость в построении опорного плана задачи линейного программирования, исходя из ее неопорного плана, встанет перед нами и при рассмотрении общего случая. В связи с этим полезно будет предварительно остановиться на способе определения опорного плана задачи VI по имеющемуся ее неопорному плану. Как будет следовать из дальнейшего, предлагаемый ниже способ естественно назвать методом градиента.

Итак, пусть $Y' = (y_1', y_2', \dots, y_m')$ — произвольный план задачи VI.

Без ограничения общности можно считать, что

$$\sum_{i=1}^m a_{ij} y_i' > c_j, \quad j = 1, 2, \dots, n_1, \quad (*)$$

$$\sum_{i=1}^m a_{ij} y_i' = c_j, \quad j = n_1 + 1, \dots, n. \quad (**)$$

В частности, n_1 может совпадать с n .

Пусть среди равенств (**) имеются r линейно независимых. Если $r = m$, Y' — опорный план задачи VI. Будем считать $r < m$.

Введем в рассмотрение систему равенств и неравенств, определяемую планом Y' ,

$$\sum_{i=1}^m a_{ij}y_i \geq c_j, \quad j=1, 2, \dots, n_1, \quad (4.63)$$

$$\sum_{i=1}^m a_{ij}y_i = c_j, \quad j=n_1+1, \dots, n. \quad (4.64)$$

Выразим r неизвестных системы (4.64) через остальные. Это можно сделать, например, последовательно их исключая. Для определенности положим, что такими неизвестными являются $y_1, y_2, \dots, y_r$

$$y_i = \sum_{j=r+1}^m d_{ij}y_j + d_i, \quad i=1, 2, \dots, r. \quad (4.65)$$

Заменяя $y_1, y_2, \dots, y_r$ в неравенствах (4.63) их выражениями из (4.65), получим систему неравенств относительно $y_{r+1}, y_{r+2}, \dots, y_m$

$$\sum_{i=r+1}^m \bar{a}_{ij}y_i \geq \bar{c}_j, \quad j=1, 2, \dots, n_1. \quad (4.66)$$

Неравенства (4.66) составляют, очевидно, необходимое и достаточное условие для того, чтобы вектор $Y = (y_1, \dots, y_m)$, компоненты которого связаны соотношениями (4.64) [или (4.65)], являлся планом задачи VI.

Подставляя в линейную форму (4.30) выражения переменных $y_1, \dots, y_r$ из (4.65), приходим к линейной функции относительно $y_{r+1}, y_{r+2}, \dots, y_m$

$$\sum_{i=r+1}^m \bar{b}_i y_i + \bar{b}_0. \quad (4.67)$$

Отправляясь теперь от вектора $Y' = (y'_1, y'_2, \dots, y'_m)$, будем двигаться по направлению наискорейшего убывания функции (4.67) (оно определяется вектором градиента этой функции, взятым с обратным знаком) до тех пор, пока хотя бы одно из неравенств системы (4.63) не обратится в равенство. Полученный план обозначим через Y'' . Аналитически поиск плана $Y'' = (y''_1, y''_2, \dots, y''_m)$ осуществляется следующим образом.

Пусть

$$y_i'(\theta) = y_i' - \theta \bar{b}_i, \quad i = r + 1, \dots, m.$$

Для того чтобы величины $y_i'(\theta)$ удовлетворяли системе неравенств (4.66), θ должно подчиняться следующим условиям:

$$\Delta_j = \sum_{i=r+1}^m \bar{a}_{ij} y_i' - \bar{c}_j \geq \theta \sum_{i=r+1}^m \bar{a}_{ij} \bar{b}_i = \theta \mu_j, \\ j = 1, 2, \dots, n_1.$$

Если все μ_j неположительны, θ можно выбрать как угодно большим (по условию $\Delta_j > 0$, $j = 1, 2, \dots, n_1$). Но тогда линейная функция (4.67) неограничена снизу на множестве переменных $y_{r+1}, \dots, y_m$, удовлетворяющих условиям (4.66). Следовательно, линейная форма задачи VI неограничена снизу на множестве ее планов. Пусть теперь среди μ_j имеются положительные. Тогда, полагая

$$\theta_0 = \min_{\mu_j > 0} \frac{\Delta_j}{\mu_j},$$

приходим к плану $Y'' = (y_1'', y_2'', \dots, y_m'')$, где $y_i'' = y_i'(\theta_0)$ для $i = r + 1, \dots, m$, а компоненты $y_1'', \dots, y_r''$ выражаются через уже определенные по формулам (4.65).

Согласно построению Y''

$$\sum_{i=1}^m a_{ij} y_i'' = c_j, \quad j = n_1 + 1, \dots, n.$$

Кроме того, если индекс j_1 определить условием $\theta_0 = \frac{\Delta_{j_1}}{\mu_{j_1}}$, то

$$\sum_{i=1}^m a_{ij_1} y_i'' = c_{j_1}.$$

Заметим, что уравнение $\sum_{i=1}^m a_{ij_1} y_i = c_{j_1}$ линейно независимо от системы уравнений (4.64). Это следует из эквивалентной формы указанной системы (4.65) и из

неравенства нулю хотя бы одного из элементов $\bar{a}_{ij_1}$, $i = r + 1, \dots, m$ ($\mu_{j_1} > 0$). Итак, план Y'' обращает в равенство по крайней мере $r + 1$ независимое условие системы (4.31). При этом

$$\sum_{i=1}^m b_i y_i'' = \sum_{j=r+1}^m \bar{b}_i y_i' + \bar{b}_0 - \theta_0 \sum_{i=r+1}^m \bar{b}_i^2 < \sum_{i=r+1}^m \bar{b}_i y_i' + \bar{b}_0 = \sum_{i=1}^m b_i y_i'.$$

До сих пор мы молчаливо предполагали линейную функцию (4.67) не равной постоянному числу.

Однако может оказаться, что все коэффициенты $\bar{b}_i$, $i = r + 1, \dots, m$ равны нулю, т. е.

$$\sum_{i=r+1}^m \bar{b}_i y_i + \bar{b}_0 = \bar{b}_0.$$

В этом случае полагаем

$$y'_{r+1}(\theta) = y'_{r+1} - \theta, \quad y_i(\theta) = y_i', \quad i = r + 2, \dots, m.$$

Тогда соответствующие условия на θ имеют вид

$$\Delta_j = \sum_{i=r+1}^m \bar{a}_{ij} y_i' - \bar{c}_j \geq \theta \bar{a}_{r+1,j}, \quad j = 1, 2, \dots, n_1.$$

Заметим, что все $\bar{a}_{r+1,j}$ не могут одновременно обратиться в нуль. Действительно, если бы это случилось, среди неравенств (4.63) не нашлось бы $m - r$ линейно независимых, что противоречит предположению о наличии m независимых условий в системе (4.31). Если среди $\bar{a}_{r+1,j}$ имеются положительные, полагаем

$$\theta_0 = \min_{\bar{a}_{r+1,j} > 0} \frac{\Delta_j}{\bar{a}_{r+1,j}}. \quad \text{В противном случае}$$

$$\theta_0 = \min_{\bar{a}_{r+1,j} < 0} \left(- \frac{\Delta_j}{\bar{a}_{r+1,j}} \right).$$

Число независимых соотношений, которым Y''' удовлетворяет как равенствам, возрастет и в этом случае. Однако линейная форма (4.30) сохранит свое прежнее значение.

Если теперь из системы (4.66) выделить все те неравенства, которым последние $m - r$ компонент Y'' удовлетворяют как равенствам, и повторить уже описанный процесс применительно к переменным $y_{r+1}, y_{r+2}, \dots, y_m$, то получим новый план Y''' , который по сравнению с Y'' обращает в равенство большее число независимых условий системы (4.31). При этом линейная форма (4.30) заведомо не возрастает.

Двигаясь таким образом от одного плана к другому, мы, в конце концов, получим опорный план задачи VI. Ясно, что при этом будет сделано не более m шагов.

Метод градиента имеет довольно наглядный геометрический смысл. Система неравенств (4.31) высекает в m -мерном пространстве переменных $y_1, y_2, \dots, y_m$ многогранное тело S — множество планов задачи VI. Каждому опорному плану соответствует вершина S . Пусть $Y' \in S$ не совпадает ни с одной из вершин S . Тогда точка Y' расположена внутри некоторой грани S_1 (S_1 может совпадать с S). Линейная форма (4.30) определяет на S_1 некоторую линейную функцию (*) [аналог (4.67)]. Метод градиента осуществляет движение от точки Y' по направлению наибоыстрейшего убывания функции (*) до пересечения с границей S_1 . Вновь полученный план Y'' лежит внутри грани S_2 , размерность которой строго меньше размерности S_1 . Понижая, таким образом, размерности граней, мы, наконец, попадаем в некоторую вершину многогранного тела S (грань нулевой размерности), соответствующую опорному плану задачи VI.

Поскольку процесс построения опорного плана методом градиента связан с монотонным убыванием линейной формы (4.30), полученный план является обычно достаточно хорошим первым приближением. Все вычисления, связанные с реализацией метода градиента, могут быть записаны в виде компактной схемы, которая основывается на методе полного исключения Гаусса (см., например, [17]), широко используемом при решении систем линейных уравнений. Оставляя в стороне общий случай, дадим описание схемы на примере конкретной задачи.

Определим методом градиента опорный план следующей задачи:

минимизировать линейную форму

$$2y_1 + 3y_2 + 3y_3 \quad (*)$$

при ограничениях:

$$\left. \begin{aligned} y_1 + 2y_2 + y_3 &\geq 1, \\ 2y_1 + y_2 + 2y_3 &\geq 1, \\ y_1 + y_2 + 2y_3 &\geq 1, \\ 2y_1 + 2y_2 + 3y_3 &\geq 2, \\ 2y_1 + 2y_2 + 2y_3 &\geq 1. \end{aligned} \right\} \quad (**)$$

В рассматриваемом случае $m=3$, $n=5$. Примем в качестве исходного плана

$$Y' = (0, 0, 1).$$

Условия задачи запишем в виде табл. 1.

Таблица 1

0	Δ	2	3	3	b c
→ 1	0	1	2	<u>1</u>	1
2	1	2	1	2	1
3	1	1	1	2	1
4	1	2	2	3	2
5	1	2	2	2	1

В столбце Δ табл. 1 выписаны разности левых и правых частей (**), образующиеся в результате подстановки $Y' = (0, 0, 1)$. Мы видим, что план Y' обращает в равенство лишь первое неравенство системы (**) ($r=1$). Поэтому необходимо из полученного уравнения выразить одну из переменных (например, y_3) и результат подставить в остальные условия задачи и линейную форму (*). Это достигается путем вычитания из строк табл. 1 строки 1, умноженной на соответствующий множитель (3 для строки 0, 2 для строки 2, и т. д.) *. Полученные результаты записываются в основную часть табл. 2.

* Если бы выделенный элемент табл. 1 отличался от единицы, следовало бы разделить на него элементы первой строки.

Таблица 2

	0	Δ'	μ'	-1	-3	0	$b' \diagup c'$
→	1	×	×	1	2	1	1
	2	$\frac{4}{13}$	9	0	-3	0	-1
	3	$\frac{3}{13}$	10	-1	-3	0	-1
→	4	0	13	$\overline{-1}$	-4	0	-1
	5	$\frac{7}{13}$	6	0	-2	0	-1

$\theta_0' = \frac{1}{13}$

После этого заполняются столбцы μ' , Δ' . Для получения μ'_i перемножаем соответствующие элементы 0-й и i -й строк табл. 2 и складываем результаты. Затем находим

$$\theta_0' = \min_{\mu'_i > 0} \left(\frac{\Delta_i}{\mu'_i} \right) = \min \left(\frac{1}{9}, \frac{1}{10}, \frac{1}{13}, \frac{1}{6} \right) = \frac{1}{13}.$$

Столбец Δ' образуется из столбца Δ табл. 1. вычитанием из него столбца μ' , каждый элемент которого предварительно умножается на $\theta_0' = \frac{1}{13}$.

В столбце Δ' имеется снова один нуль. Выделяем содержащую его строку табл. 2 (строка 4) и повторяем всю последовательность операций. Приходим к табл. 3.

Таблица 3

	0	Δ''	μ''	0	1	0	$b'' \diagup c''$
→	1	×	×	0	-2	1	0
	2	1	-3	0	-3	0	-1
→	3	0	1	0	$\overline{1}$	0	0
→	4	×	×	1	4	0	1
	5	1	-2	0	-2	0	-1

$\theta_0'' = \frac{3}{13}$

В результате выделенными оказываются 3 строки таблицы. Отправляясь от последней из выделенных строк (строка 3), заполняем заключительную табл. 4.

Таблица 4

	0	Δ^m	μ^m	0	0	0	b^m / c^m
→	1	×	×	0	0	1	0
	2			0	0	0	-1
→	3	×	×	0	1	0	0
→	4	×	×	1	0	0	1
	5			0	0	0	-1

Первые два столбца этой таблицы остаются незаполненными. В каждой из строк табл. 4 (в пределах столбцов 3, 4, 5) либо одни нули, либо единственная 1.

Компоненты опорного плана $Y = (y_1, y_2, y_3)$ определяются столбцом c''' (столбец 6). Для получения y_i следует двигаться вдоль $(i+2)$ -го столбца табл. 4 до 1. Элемент столбца c''' , расположенный в строке этой единицы, совпадает с искомой компонентой

$$y_1 = 1; y_2 = 0; y_3 = 0.$$

Предоставим читателю убедиться, что построенный опорный план $Y = (1, 0, 0)$ является решением задачи (*), (**). Метод градиента может быть применен лишь в тех случаях, когда известен некоторый план сопряженной задачи. Как уже отмечалось, для широкого класса задач определение плана (не обязательно опорного) осуществляется очень просто. Тем не менее в некоторых случаях отыскать такой план довольно затруднительно. В подобных ситуациях можно воспользоваться приемом, излагаемым ниже. Дополним ограничения (4.28) задачи V условием

$$x_1 + x_2 + \dots + x_n \leq M$$

или

$$x_0 + x_1 + x_2 + \dots + x_n = M, x_0 \geq 0. \quad (4.68)$$

Вновь полученную задачу (с переменными $x_0, x_1, \dots, x_n$) обозначим через V' . Задача, сопряженная с V' , состоит в отыскании минимума линейной формы

$$My_0 + \sum_{i=1}^m b_i y_i, \quad (4.69)$$

переменные которой подчинены ограничениям

$$y_0 + \sum_{i=1}^m a_{ij} y_i \geq c_j, \quad j = 1, 2, \dots, n; \quad (4.70)$$

$$y_0 \geq 0.$$

Обозначим эту задачу через VI' . Очевидно, в качестве плана задачи VI' можно взять

$$Y' = (y_0, 0, 0, \dots, 0),$$

где

$$y_0 = \max_j \{c_j, 0\}.$$

Применив далее метод градиента, получим опорный план этой задачи. Теперь мы в состоянии решить задачу V' методом последовательного уточнения оценок. В процессе решения величина M считается достаточно большой (больше любого числа, с которым по ходу решения ее необходимо сравнивать). Допустим, что оптимальные планы задач V' , VI' уже определены:

$$\bar{X}^* = (x_0^*, x_1^*, \dots, x_n^*),$$

$$\bar{Y}^* = (y_0^*, y_1^*, \dots, y_m^*).$$

Возможны два случая: а) $y_0^* = 0$; б) $y_0^* > 0$.

Если имеет место случай а), $Y^* = (y_1^*, \dots, y_m^*)$ является, очевидно, решением задачи VI . Поэтому $X^* = (x_1^*, \dots, x_n^*)$ — оптимальный план задачи V

$$\left(\sum_{j=1}^n c_j x_j^* + 0 \cdot x_0^* = \sum_{i=1}^m b_i y_i^* + M \cdot 0 \right).$$

Во втором случае б) множество планов задачи VI пусто. Действительно, пусть задача VI имеет хотя бы один план $Y = (y_1, \dots, y_m)$. Тогда $\bar{Y} = (0, y_1, y_2, \dots, y_m)$ — план задачи VI' , причем при достаточно большом M значение линейной формы (4.69) в точке

$\bar{Y}^* = (y_0^*, y_1^*, \dots, y_m^*)$ превосходит ее значение в точке $\bar{Y} = (0, y_1, y_2, \dots, y_m)$, что противоречит оптимальности плана $\bar{Y}^*$. Итак, при $y_0^* > 0$ задача VI, а следовательно, и сопряженная с ней задача V неразрешимы.

Таким образом, в случае разрешимости исследуемой задачи рассмотренный прием позволяет найти ее оптимальный план без предварительного определения плана сопряженной задачи.

2.6. Случай вырожденности. До сих пор сопряженная задача VI предполагалась невырожденной. По определению это означает, что каждый ее опорный план обращает в равенства ровно m соотношений системы (4.31). Попытаемся теперь освободиться от этого ограничительного допущения.

Прежде всего следует указать на те затруднения, которые возникают в вырожденном случае при реализации метода последовательного уточнения оценок. Если проанализировать рассматриваемый метод, то окажется, что невырожденность сопряженной задачи была использована нами в двух местах:

1) при установлении монотонности процесса решения, которая следует из θ_0 положительности

$$\theta_0 = \min_{z_{kj} < 0} \left(-\frac{\Delta_j}{z_{kj}} \right);$$

2) при доказательстве однозначности выбора вектора A_l из условия

$$\theta_0 = -\frac{\Delta_l}{z_{kl}}.$$

Если сопряженная задача VI вырожденная, некоторые из $\Delta_j = \sum_{i=1}^m a_{ij}y_i - c_j$, для которых $z_{kj} < 0$, могут оказаться равными нулю. Тогда

$$\theta_0 = \min_{z_{kj} < 0} \left(-\frac{\Delta_j}{z_{kj}} \right) = 0,$$

и, следовательно, построение нового плана задачи VI не приведет к уменьшению ее линейной формы. Вспомним, что конечность метода была выведена из невозможности возврата к пройденному ранее базису; последняя же являлась следствием монотонного убывания линейной формы задачи VI. Таким образом, в вырожденном случае возврат к старому базису (или, как говорят, цикл) вполне возможен. Условия, которые использовались ранее для выделения (однозначного) вектора, подлежащего вводу в новый базис, оказываются теперь, вообще говоря, недостаточными; им могут удовлетворять сразу несколько векторов из системы $\{A_j\}$, $j = m + 1, \dots, n$. Если из этих векторов выбирать любой, то не исключено получение цикла — периодического возвращения через определенное число итераций к одному и тому же базису. Пример такого цикла был построен Биллом [22]. Ясно, что при наличии цикла процесс решения никогда не будет закончен. Поэтому необходимо несколько дополнить метод последовательного уточнения оценок с тем, чтобы исключить возможность заикливания. Для этого может быть использован все тот же ε -прием (§ 7 гл. 3), состоящий в незначительном изменении компонент вектора ограничений. Как обычно, будем предполагать, что исходный план сопряженной задачи VI обладает базисом, состоящим из векторов $A_1, A_2, \dots, A_m$.

Задавшись $\varepsilon > 0$, рассмотрим задачу линейного программирования, состоящую в минимизации формы (4.30), переменные которой удовлетворяют условиям

$$\sum_{i=1}^m a_{ij}y_i \geq c_j - \varepsilon^{n+1-j} = c'_j(\varepsilon), \quad j = 1, 2, \dots, n. \quad (4.31')$$

Таким образом, каждое из неравенств системы (4.31') отличается от соответствующего неравенства системы (4.31) наличием в правой части слагаемого $-\varepsilon^{n+1-j}$, где j — номер рассматриваемого условия. Покажем, что при достаточно малом ε задача (4.30), (4.31') — невырожденная.

Для этого рассмотрим произвольную систему из $(m + 1)$ расширенных векторов условий задачи (4.30), (4.31') $\bar{A}_{j_1}, \bar{A}_{j_2}, \dots, \bar{A}_{j_{m+1}}$, из которых m — линейно

независимы. Составим из координат этих векторов определитель

$$\Delta_{j_1, j_2, \dots, j_{m+1}}(\epsilon) = \begin{vmatrix} a_{1j_1} & a_{1j_2} \dots a_{1j_{m+1}} \\ a_{2j_1} & a_{2j_2} \dots a_{2j_{m+1}} \\ \vdots & \vdots \dots \vdots \\ a_{mj_1} & a_{mj_2} \dots a_{mj_{m+1}} \\ c'_{j_1}(\epsilon) & c'_{j_2}(\epsilon) \dots c'_{j_{m+1}}(\epsilon) \end{vmatrix}$$

порядка $m + 1$. Пользуясь элементарными свойствами определителей, получаем

$$\Delta_{j_1, j_2, \dots, j_{m+1}}(\epsilon) = \sum_{s=1}^{m+1} \alpha_s \epsilon^{n+1-j_s} + \alpha_0,$$

где

$$\alpha_s = (-1)^{m+s} \begin{vmatrix} a_{1j_1} & a_{1j_2} \dots a_{1j_{s-1}} & a_{1j_{s+1}} \dots a_{1j_{m+1}} \\ a_{2j_1} & a_{2j_2} \dots a_{2j_{s-1}} & a_{2j_{s+1}} \dots a_{2j_{m+1}} \\ \dots & \dots & \dots \\ a_{mj_1} & a_{mj_2} \dots a_{mj_{s-1}} & a_{mj_{s+1}} \dots a_{mj_{m+1}} \end{vmatrix},$$

$$s = 1, 2, \dots, m + 1;$$

$$\alpha_0 = \begin{vmatrix} a_{1j_1} & a_{1j_2} \dots a_{1j_{m+1}} \\ a_{2j_1} & a_{2j_2} \dots a_{2j_{m+1}} \\ \dots & \dots \\ a_{mj_1} & a_{mj_2} \dots a_{mj_{m+1}} \\ c_{j_1} & c_{j_2} \dots c_{j_{m+1}} \end{vmatrix}.$$

Поскольку среди векторов $A_{j_1}, A_{j_2}, \dots, A_{j_{m+1}}$ имеются m линейно независимых, хотя бы один из определителей $\alpha_s, s = 1, 2, \dots, m + 1$, отличен от нуля.

Поэтому многочлен $P(\epsilon)^{(j_1, j_2, \dots, j_{m+1})} = \sum_{s=1}^{m+1} \alpha_s \epsilon^{n+1-j_s} + \alpha_0$ не может быть тождественным нулем,

Обозначим через $\nu_1, \nu_2, \dots, \nu_k$ его корни, число которых, очевидно, не превышает n . Пусть

$$\nu(j_1, j_2, \dots, j_{m+1}) = \min_{\nu_i > 0} \nu_i,$$

$(\nu(j_1, \dots, j_{m+1})$ — минимальный положительный корень многочлена $P(\epsilon)^{(j_1, \dots, j_{m+1})}$. Если положительных корней нет, то $\nu(j_1, \dots, j_{m+1})$ полагается равным ∞ . Определим теперь число ϵ' условием

$$\epsilon' = \min \nu(j_1, \dots, j_{m+1}),$$

где минимум берется по всевозможным системам $(A_{j_1}, A_{j_2}, \dots, A_{j_{m+1}})$ содержащим m линейно независимых векторов. Поскольку число таких систем конечно (не превосходит C_n^{m+1}), $\epsilon' > 0$. Проверим, что при любом ϵ , удовлетворяющем условию

$$0 < \epsilon < \epsilon', \quad (4.71)$$

задача (4.30), (4.31') невырожденная.

Пусть $Y = (y_1, \dots, y_m)$ — произвольный опорный план этой задачи с базисом $A_{j_1}, A_{j_2}, \dots, A_{j_m}$. Тогда

$$\sum_{i=1}^m a_{ij} y_i > c_j' = c_j - \epsilon^{n+1-j} \quad (4.72)$$

для всех $j \neq j_s, s = 1, 2, \dots, m$.

В самом деле, предположив противное, мы получили бы систему равенств

$$\sum_{i=1}^m a_{ij_s} y_i = c_{j_s} - \epsilon^{n+1-j_s}, \quad s = 1, 2, \dots, m, m+1,$$

где $j_{m+1} \neq j_s, s = 1, 2, \dots, m$.

Откуда, пользуясь условием разрешимости систем линейных уравнений, имеем

$$\Delta_{j_1, j_2, \dots, j_{m+1}}(\epsilon) = 0,$$

что противоречит условию (4.71). Таким образом, система неравенств (4.72) действительно имеет место. Для завершения доказательства невырожденности задачи (4.30), (4.31') остается сослаться на произвольность опорного плана $Y = (y_1, \dots, y_m)$. Наряду с исходной задачей (4.27), (4.28), (4.29) (V) рассмотрим задачу

(4.27'), (4.28), (4.29), где линейная функция (4.27') имеет вид

$$\sum_{j=1}^n c_j'(\epsilon) x_j = \sum_{j=1}^n (c_j - \epsilon^{n+1-j}) x_j. \quad (4.27')$$

Установим существование такого $\epsilon'' > 0$, что при любом ϵ , удовлетворяющем условию

$$0 < \epsilon < \epsilon'', \quad (4.73)$$

каждое опорное решение задачи (4.27'), (4.28), (4.29) является вместе с тем решением задачи V.

Пусть X^* — решение задачи V и

$$m = \min \left[\sum_{j=1}^n c_j (x_j^* - x_j) \right],$$

где минимум берется по всевозможным ее опорным, неоптимальным планам. Поскольку число опорных планов задачи V конечно, $m > 0$. Выберем ϵ'' из условия

$$\sum_{j=1}^n (\epsilon'')^{n+1-j} x_j^* < \frac{m}{2}. \quad (4.74)$$

Рассмотрим теперь произвольное опорное решение задачи (4.27'), (4.28), (4.29) $X_\epsilon = (x_1^{(\epsilon)}, \dots, x_n^{(\epsilon)})$. Если бы план X_ϵ не являлся оптимальным для задачи V, то

$$\sum_{j=1}^n c_j x_j^* - m \geq \sum_{j=1}^n c_j x_j^{(\epsilon)} > \sum_{j=1}^n (c_j - \epsilon^{n+1-j}) x_j^{(\epsilon)}. \quad (4.75)$$

С другой стороны, из (4.74) получаем

$$\sum_{j=1}^n (c_j - \epsilon^{n+1-j}) x_j^* > \sum_{j=1}^n c_j x_j^* - \frac{m}{2}. \quad (4.76)$$

Сравнивая (4.75) и (4.76), приходим к неравенству

$$\sum_{j=1}^n (c_j - \epsilon^{n+1-j}) x_j^* > \sum_{j=1}^n (c_j - \epsilon^{n+1-j}) x_j^{(\epsilon)},$$

которое противоречит оптимальности X_ϵ для задачи (4.27'), (4.28), (4.29). Итак, X_ϵ — решение исходной задачи V.

Покажем, наконец, что векторы $A_1, A_2, \dots, A_m$, составляющие по условию базис исходного плана $Y = (y_1, \dots, y_m)$ задачи VI, образуют также базис для некоторого плана задачи (4.30), (4.31').

Пусть величины $y_1^{(\epsilon)}, y_2^{(\epsilon)}, \dots, y_m^{(\epsilon)}$ являются решением системы линейных уравнений

$$\sum_{i=1}^m a_{ij} y_i^{(\epsilon)} = c_j'(\epsilon) = c_j - \epsilon^{n+1-j}, \quad j = 1, 2, \dots, m. \quad (4.77)$$

По условию компоненты плана $Y = (y_1, \dots, y_m)$ удовлетворяют системе уравнений

$$\sum_{i=1}^m a_{ij} y_i = c_j, \quad j = 1, 2, \dots, m. \quad (4.78)$$

Вычитая из уравнений системы (4.78) соответствующие уравнения системы (4.77) и полагая $\Delta y_i = y_i - y_i^{(\epsilon)}$, $i = 1, 2, \dots, m$, получаем

$$\sum_{i=1}^m a_{ij} \Delta y_i = \epsilon^{n+1-j}, \quad j = 1, 2, \dots, m. \quad (4.79)$$

Используя далее правило Крамера для решения систем линейных уравнений, нетрудно установить следующие оценки:

$$|\Delta y_i| \leq c \epsilon^{n+1-m}, \quad i = 1, 2, \dots, m, \quad (4.80)$$

где c не зависит от ϵ .

Проверим теперь, пользуясь неравенством (4.80), что при достаточно малом ϵ $Y_\epsilon = (y_1^{(\epsilon)}, \dots, y_m^{(\epsilon)})$ является планом задачи (4.30), (4.31'). В самом деле, для любого $m+1 \leq j \leq n$

$$\begin{aligned} \sum_{i=1}^m a_{ij} y_i^{(\epsilon)} &= \sum_{i=1}^m a_{ij} (y_i - \Delta y_i) \geq c_j - c \epsilon^{n+1-m} \sum_{i=1}^m |a_{ij}| = \\ &= c_j - c^* \epsilon^{n+1-m}, \end{aligned}$$

где

$$c^* = c \max_{m+1 \leq j \leq n} \sum_{i=1}^m |a_{ij}|.$$

Положим $\epsilon''' = \frac{1}{c^*}$. Очевидно, что при $0 < \epsilon < \epsilon'''$ (4.81)

$$\epsilon^{n+1-j} > c^* \epsilon^{n+1-m} \text{ для } j = m+1, m+2, \dots, n.$$

Поэтому

$$\sum_{i=1}^m a_{ij} y_i^{(\epsilon)} \geq c_j - c^* \epsilon^{n+1-m} > c_j - \epsilon^{n+1-j},$$

$$j = m+1, m+2, \dots, n. \quad (4.82)$$

Из соотношений (4.77), (4.82) следует, что $Y_\epsilon = (y_1^{(\epsilon)}, \dots, y_m^{(\epsilon)})$ — опорный план задачи (4.30), (4.31') с базисом $A_1, A_2, \dots, A_m$. Теперь уже нетрудно обосновать прием, позволяющий использовать метод последовательного уточнения оценок как в невырожденном, так и в вырожденном случае. Применим к задаче (4.27'), (4.28), (4.29), в которой

$$0 < \epsilon < \epsilon_0 = \min \{\epsilon', \epsilon'', \epsilon'''\}, \quad (4.83)$$

метод уточнения оценок. Допустим, что базис исходного плана задачи VI составляется из векторов $A_1, A_2, \dots, A_m$ (этого всегда можно добиться соответствующей перенумерацией векторов условий задачи V). Тогда согласно неравенствам (4.81), (4.83) те же векторы образуют базис и для некоторого плана задачи (4.30), (4.31'), который мы примем за исходный. В силу условий (4.71), (4.83) сопряженная задача (4.30), (4.31') невырожденная и, следовательно, изучаемый метод через конечное число итераций приведет к решению задачи (4.27'), (4.28), (4.29). В соответствии же с условиями (4.73), (4.83) полученное решение будет вместе с тем решением исходной задачи V. Заметим, что при практическом использовании описанного приема нет нужды проверять справедливость неравенства (4.83). Достаточно лишь считать ϵ меньшим любого из положительных чисел, с которыми в ходе решения задачи его необходимо сравнивать. Как и в методе последовательного улучшения плана, ϵ -прием позволяет сформулировать правило однозначного выбора вводимого в базис вектора, исключающее возможность заикливания.

Пусть $Y_\epsilon = (y_1^{(\epsilon)}, y_2^{(\epsilon)}, \dots, y_m^{(\epsilon)})$ — опорный план задачи (4.30), (4.31') с базисом $A_1, A_2, \dots, A_m$,

Используя систему уравнений (4.77), которой удовлетворяют компоненты плана Y_ϵ , а также определение величин e_{st} , $s=1, 2, \dots, m$; $t=1, 2, \dots, m$ [см. п. 2.2, (4.44)] получаем

$$y_i^{(\epsilon)} = y_i - \sum_{v=1}^m \epsilon^{n+1-v} e_{vi}, \quad (4.84)$$

где $Y = (y_1, y_2, \dots, y_m)$ является планом задачи VI с тем же базисом $A_1, A_2, \dots, A_m$.

В соответствии с рассматриваемым методом условия ввода в новый базис вектора A_l состоят в равенстве

$$-\frac{\Delta_l^{(\epsilon)}}{z_{kl}} = \min_{z_{kj} < 0} \left(-\frac{\Delta_j^{(\epsilon)}}{z_{kj}} \right),$$

где

$$\Delta_j^{(\epsilon)} = \sum_{i=1}^m a_{ij} y_i^{(\epsilon)} - (c_j - \epsilon^{n+1-j}); \quad z_{kj} = \sum_{i=1}^m a_{ij} e_{ki}.$$

Воспользовавшись равенством (4.84), имеем

$$\begin{aligned} \Delta_j^{(\epsilon)} &= \sum_{i=1}^m a_{ij} y_i^{(\epsilon)} - (c_j - \epsilon^{n+1-j}) = \\ &= \sum_{i=1}^m a_{ij} y_i - \sum_{i=1}^m a_{ij} \sum_{v=1}^m \epsilon^{n+1-v} e_{vi} - c_j + \epsilon^{n+1-j}. \end{aligned}$$

Преобразуем полученную двойную сумму

$$\sum_{i=1}^m a_{ij} \sum_{v=1}^m \epsilon^{n+1-v} e_{vi} = \sum_{v=1}^m \epsilon^{n+1-v} \sum_{i=1}^m a_{ij} e_{vi} = \sum_{v=1}^m \epsilon^{n+1-v} z_{vj}.$$

Итак,

$$\begin{aligned} \Delta_j^{(\epsilon)} &= \sum_{i=1}^m a_{ij} y_i - \sum_{v=1}^m \epsilon^{n+1-v} z_{vj} + \epsilon^{n+1-j} - c_j = \\ &= \Delta_j - \sum_{v=1}^m \epsilon^{n+1-v} z_{vj} + \epsilon^{n+1-j}. \end{aligned}$$

Откуда

$$-\frac{\Delta_j^{(\epsilon)}}{z_{kj}} = -\frac{\Delta_j}{z_{kj}} - \frac{\epsilon^{n+1-j}}{z_{kj}} + \sum_{v=1}^m \frac{z_{vj}}{z_{kj}} \epsilon^{n+1-v}, \quad (4.85)$$

$$j = m+1, m+2, \dots, n.$$

Из полученного соотношения легко выводится искомое правило образования нового базиса. Прежде всего перенумеруем векторы условий задачи V так, чтобы базис исходного плана сопряженной задачи VI состоял из векторов $A_1, A_2, \dots, A_m$. В процессе решения установленную нумерацию следует сохранять. Пусть после нескольких итераций мы пришли к базису $A_{j_2}, A_{j_2}, \dots, A_{j_m}$, причем оказалось, что из него следует вывести вектор A_{j_k} . Отыскание вектора, подлежащего вводу в базис, проводится следующим образом. Определяем $\min_{z_{jkj} < 0} \left\{ -\frac{\Delta_j}{z_{jkj}} \right\}$. Если этот минимум достигается

лишь на одной из сравниваемых величин, скажем на $-\frac{\Delta_l}{z_{jkl}}$, в базис вводится вектор A_l . Если же сразу на нескольких величинах, например на двух с номерами l_1, l_2 , производится выбор между векторами A_{l_1}, A_{l_2} . Будем предполагать, что числа $j_1, j_2, \dots, j_m$ расположены в порядке возрастания. Прежде всего отбросим те из „подозрительных“ векторов, номера которых больше j_m (если этому условию удовлетворяют все подобные векторы, в базис вводится „подозрительный“ вектор с наименьшим номером). Если в результате останется лишь один „подозрительный“ вектор, он вводится в базис. Если же $l_1, l_2 < j_m$, то определяется

$$\min \left\{ \frac{z_{jml_1}}{z_{jkl_1}}, \frac{z_{jml_2}}{z_{jkl_2}} \right\}.$$

Возможны два случая:

а) минимум достигается на одной из сравниваемых величин, скажем на первой. Тогда в базис вводится вектор A_{l_1} ;

б) обе величины (несколько величин) равны. В этом случае поиск вектора, подлежащего вводу в базис, продолжается: оставшиеся «подозрительные» векторы подвергаются уже описанным операциям с той разницей, что вектор A_{j_m} заменяется на $A_{j_{m-1}}$.

Если вводимый вектор и теперь не будет определен однозначно, процесс поиска, связанный последовательно с векторами $A_{j_{m-2}}, A_{j_{m-3}}, \dots, A_{j_1}$ продол-

жается. Назовем шагом процесса поиска совокупность операций, связанных с одним из векторов старого базиса $(A_{j_1}, A_{j_2}, \dots, A_{j_m})$. Чтобы отыскать вектор, подлежащий вводу в новый базис, потребуется, очевидно, не более чем m таких шагов. Для обоснования описанного правила достаточно заметить, что согласно (4.85) оно приводит при малых ε к отысканию минимальной из величин $-\frac{\Delta_j^{(\varepsilon)}}{z_{jkj}}$, для которых $z_{jkj} < 0$ (то, что формула (4.85) получена для базиса $A_1, A_2, \dots, A_m$, нисколько не отражается на ее общности).

Таким образом, метод последовательного уточнения оценок с одинаковым успехом может применяться как в невырожденном, так и в вырожденном случае. Вырожденность сопряженной задачи ведет лишь иногда к некоторому замедлению выбора вводимого вектора. Что же касается числа итераций, то, несмотря на отсутствие, вообще говоря, монотонного убывания линейной формы (4.30), оно всегда конечно. Следует отметить, что при решении практических задач рассматриваемым методом, равно как и методом улучшения плана, получение цикла маловероятно. Более того, построение примеров заикливания требует известной изощренности. Вместе с тем изложенный только что способ, полностью гарантирующий от возможности заикливания, сравнительно сложен. Поэтому для однозначного выбора вектора, подлежащего вводу в новый базис, обычно пользуются более простыми правилами. Одно из таких правил, например, состоит в следующем: в базис вводится «подозрительный» вектор с наименьшим (или наибольшим) порядковым номером. В подавляющем большинстве случаев указанное элементарное правило делает невозможным возврат к уже раз пройденному базису. Если же цикл будет все-таки получен, необходимо, начиная от пройденного дважды базиса, использовать общее правило, описанное в этом пункте. Им следует руководствоваться до первого уменьшения линейной формы сопряженной задачи, после чего целесообразно вернуться к одному из упрощенных правил.

2.7. Методы улучшения плана и уточнения оценок. Если сравнить метод последовательного улучшения плана (II алгоритм) и рассмотренный только что метод по-

следовательного уточнения оценок, то окажется, что их вычислительные схемы довольно сходны. В обоих методах процесс решения заключается в последовательном заполнении основных таблиц и продолжении на каждом шаге вспомогательной табл. II. Отдельная итерация каждого из методов состоит в переходе от одной линейно независимой системы m векторов условий к другой. В методе улучшения плана все такие системы являются базисами исходной задачи, в методе уточнения оценок они составляют базисы сопряженной задачи. Как в том, так и в другом методе процесс решения завершается получением системы векторов $A_{i_1}, A_{i_2}, \dots, A_{i_m}$, служащей базисом для некоторых планов исходной и сопряженной задач. Что касается пары планов, соответствующих этому базису, то каждый из них является решением своей задачи. Укажем на одно обстоятельство, отличающее сравниваемые методы друг от друга. В методе уточнения оценок указывается, какой вектор следует вывести из базиса. Вводимый вектор, вообще говоря, определяется однозначно лишь в случае невырожденности сопряженной задачи. В вырожденном случае для выбора этого вектора иногда необходимо привлекать дополнительные соображения. В методе улучшения плана имеет место противоположная ситуация: вводимый вектор указывается точно, для определения же выводимого вектора в некоторых случаях необходимо прибегать к особым правилам*. Если попытаться оценить оба метода по их трудоемкости, то в общем случае ни одному из них нельзя отдать предпочтения. В некоторых задачах выгоднее использовать метод улучшения плана, в других — быстрее к цели ведет метод уточнения оценок. Поэтому выбор метода следует осуществлять для каждой конкретной задачи (или класса задач).

Например, в ряде случаев отыскание первоначального плана исходной задачи связано со значительными трудностями, тогда как один из планов сопряженной задачи определяется без всяких выкладок. В подобной ситуации

* При описании метода улучшения плана через A_k обозначался вводимый в базис вектор. В методе уточнения оценок индекс k имел вектор, исключаемый из базиса. Таким образом, в обоих методах через A_k обозначался вектор, определяемый однозначно во всех случаях (он отыскивается в начале итерации).

естественно отдать предпочтение методу уточнения оценок.

Отметим одно полезное свойство метода уточнения оценок. Иногда после решения задачи линейного программирования необходимо бывает ответить на следующий вопрос: как изменится полученный оптимальный план, если к условиям задачи добавить одно или несколько новых ограничений. Рассмотрим, например, известную задачу транспортировки (§ 10 гл. 2), решение которой состоит в составлении наиболее экономного плана перевозок. Естественно, что каждая из коммуникаций, связывающих пункты задачи, имеет определенную пропускную способность. В первоначальном решении транспортной задачи это обстоятельство обычно не учитывается (пропускные способности со временем могут меняться). Поэтому после отыскания оптимального плана перевозок может оказаться, что величины грузопотоков по некоторым из коммуникаций превышают их пропускные способности. В этом случае план перевозок необходимо пересмотреть. Можно было бы, конечно, не используя имеющегося плана, решить задачу заново с учетом необходимых ограничений. Однако подобный путь вряд ли следует признать целесообразным, поскольку при его реализации усилия, затраченные на определение старого плана, оказались бы бесполезными. Возникает необходимость разумного учета полезной информации, заключенной в уже имеющемся плане. Для этой цели с успехом может быть использован метод уточнения оценок. Обратимся к общему случаю. Пусть в результате решения задачи V получен оптимальный план $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ с базисом $A_{i_1}, A_{i_2}, \dots, A_{i_m}$. Поставим теперь новую задачу линейного программирования, образованную из V путем добавления ограничения

$$\sum_{i=1}^n d_i x_i \leq b_{m+1},$$

или, что то же самое,

$$\sum_{i=1}^n d_i x_i + x_{n+1} = b_{m+1}, \quad x_{n+1} \geq 0.$$

Назовем эту задачу V_1 . Используя обычные правила, сформулируем задачу VI_1 , сопряженную V_1 : минимизировать линейную форму

$$\sum_{i=1}^{m+1} b_i y_i$$

при условиях

$$\sum_{i=1}^m a_{ij} y_i + d_j y_{m+1} \geq c_j, \quad j = 1, 2, \dots, n,$$

$$y_{m+1} \geq 0.$$

Если

$$\sum_{i=1}^n d_i x_i^* \leq b_{m+1},$$

то вектор $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ является, очевидно, искомым решением задачи V_1 .

Пусть теперь

$$\sum_{i=1}^n d_i x_i^* > b_{m+1}.$$

В таком случае применим для решения усложненной задачи (V_1) метод уточнения оценок.

Если задача V исследовалась с помощью метода улучшения плана (II алгоритм) или метода уточнения оценок, то в последней основной таблице помещаются:

- а) оптимальный план задачи $V — X^* = (x_1^*, x_2^*, \dots, x_n^*)$;
- б) оптимальный план задачи $VI — Y^* = (y_1^*, y_2^*, \dots, y_m^*)$;
- в) коэффициенты разложения единичных векторов $e_1, e_2, \dots, e_m$ по базису $A_{j_1}, \dots, A_{j_m} — \|e_{st}^*\|$, составляющие матрицу, обратную матрице $(A_{j_1}, A_{j_2}, \dots, A_{j_m})$.

Из условий задачи VI_1 следует, что вектор $Y^* = (y_1^*, \dots, y_m^*, 0)$ является ее опорным планом с базисом $\bar{A}_{j_1}, \bar{A}_{j_2}, \dots, \bar{A}_{j_m}, \bar{A}_{n+1}$. Здесь по определению

$$\bar{A}_j = (a_{1j}, a_{2j}, \dots, a_{mj}, d_j), \quad j = 1, 2, \dots, n;$$

$$\bar{A}_{n+1} = (0, 0, \dots, 0, 1).$$

Следовательно, при решении задачи V_1 методом уточнения оценок этот вектор можно принять за исходный план

задачи VI₁. Заметим, что соответствующая ему основная таблица без труда образуется из последней основной таблицы задачи V:

$$\text{а) } \bar{X}^* = (x_1^*, x_2^*, \dots, x_n^*, x_{n+1}^*),$$

где

$$x_{n+1}^* = - \sum_{j=1}^m d_j x_j^* + b_{m+1};$$

$$\text{б) } \bar{Y}^* = (y_1^*, y_2^*, \dots, y_m^*, 0);$$

$$\text{в) } \bar{e}_{st}^* = e_{st}^*, \quad s, t = 1, 2, \dots, m;$$

$$\bar{e}_{s, m+1}^* = 0, \quad s = 1, 2, \dots, m;$$

$$\bar{e}_{m+1, t}^* = - \sum_{\lambda=1}^m e_{\lambda t}^* d_{\lambda}, \quad t = 1, 2, \dots, m;$$

$$\bar{e}_{m+1, m+1}^* = 1.$$

Справедливость выписанных соотношений предоставляем проверить читателю.

Обычно для получения плана усложненной задачи требуется небольшое число итераций метода уточнения оценок. Объясняется это тем, что план $\bar{Y}^*$ почти всегда оказывается близким к решению задачи VI₁.

В заключение укажем на возможность совместного использования обоих сравниваемых методов. При решении задачи линейного программирования методом улучшения плана полезно уметь оценивать проигрыш в ее линейной форме, возникающий за счет замены оптимального плана одним из промежуточных. Имея подобные оценки, можно обрывать процесс решения задачи, как только значение ее линейной формы достаточно приблизится к оптимуму. Во многих случаях это позволяет сократить вычислительную работу. Рассмотрим задачу линейного программирования V и сопряженную с ней задачу VI. Пусть $X = (x_1, x_2, \dots, x_n)$ и $Y = (y_1, y_2, \dots, y_m)$ — некоторые опорные планы этих задач. Если через V обозначить оптимум исходной задачи V то, как известно,

$$V \leq \sum_{i=1}^m b_i y_i.$$

Следовательно,

$$V - \sum_{j=1}^n c_j x_j \leq \sum_{i=1}^m b_i y_i - \sum_{j=1}^n c_j x_j. \quad (4.86)$$

Таким образом, обладая некоторым планом сопряженной задачи, можно оценить отклонение произвольного плана исходной задачи от оптимального. В соответствии с первой теоремой двойственности оптимальные значения линейных форм исходной и сопряженной задач совпадают. Поэтому оценка (4.86) будет тем точнее, чем ближе Y к решению задачи VI. Допустим теперь, что к задаче V применяются одновременно оба метода: метод улучшения плана, начиная с плана X , и метод уточнения оценок, начиная с плана Y . В результате образуются две последовательности планов:

$$X, X_1, X_2, \dots, X_k; Y, Y_1, Y_2, \dots, Y_k.$$

Используя оценку (4.86), получаем

$$\Delta_k = V - \sum_{j=1}^n c_j x_j^{(k)} \leq \sum_{i=1}^m b_i y_i^{(k)} - \sum_{j=1}^n c_j x_j^{(k)} = \bar{\Delta}_k.$$

Если величина $\bar{\Delta}_k$ укладывается в заранее заданные пределы, процесс решения прекращается и в качестве оптимального берется план X_k . В противном случае осуществляется переход к новой паре планов X_{k+1}, Y_{k+1} . Рассмотренное совмещение метода улучшения плана и метода уточнения оценок, позволяющее каждому промежуточному плану исходной задачи сопоставить оценку его близости к искомому решению, принято называть комбинированным методом. Этот метод особенно эффективен в применении к задачам с большим числом переменных и ограничений.

2.8. Пример. Максимизировать с помощью метода уточнения оценок линейную форму

$$L(x) = x_1 + 2x_2 + 3x_3 + x_4 + 2x_5 + 3x_6 + x_7$$

при условиях:

$$\begin{aligned} x_1 + x_2 + 2x_3 + 3x_4 + 2x_5 + 3x_6 + x_7 &= 7, \\ 2x_1 + 3x_2 + x_3 + x_4 + 3x_5 + 2x_6 + 2x_7 &= 8, \\ x_1 + 2x_2 + 3x_3 + 2x_4 + 0,5x_5 + x_6 + x_7 &= 6, \\ 2x_1 + x_2 + 3x_3 + x_4 + 2x_5 + 3x_6 + x_7 &= 7, \\ x_j &\geq 0, j = 1, 2, 3, 4, 5, 6, 7. \end{aligned}$$

Прежде всего выпишем соответствующую двойственную задачу.

Минимизировать линейную форму

$$7y_1 + 8y_2 + 6y_3 + 7y_4$$

при условиях:

$$\begin{aligned} y_1 + 2y_2 + y_3 + 2y_4 &\geq 1, \\ y_1 + 3y_2 + 2y_3 + y_4 &\geq 2, \\ 2y_1 + y_2 + 3y_3 + 3y_4 &\geq 3, \\ 3y_1 + y_2 + 2y_3 + y_4 &\geq 1, \\ 2y_1 + 3y_2 + 0,5y_3 + 2y_4 &\geq 2, \\ 3y_1 + 2y_2 + y_3 + 3y_4 &\geq 3, \\ y_1 + 2y_2 + y_3 + y_4 &\geq 1. \end{aligned}$$

Для начала процесса необходимо иметь опорный план этой задачи. Определим его с помощью метода градиента (п. 2.5), отправляясь от неопорного плана $Y = (0, 0, 4, 0)$. Вычисления проводятся по схеме, изложенной в п. 2.5, в которую внесены некоторые добавления. Чтобы использовать метод уточнения оценок, необходимо вместе с опорным планом сопряженной задачи получить связанную с ним матрицу $\|e_{st}\|$, составляющую главную часть основной табл. Ia. Элементы этой матрицы можно было бы вычислить, решив системы уравнений (4.44), определяемые базисом опорного плана. Метод градиента, как мы видели, позволяет: а) отыскать базис опорного плана, векторы которого определяют матрицу коэффициентов систем (4.44), б) найти опорный план $\bar{Y}$, компоненты которого составляют решение системы

$$\sum_{i=1}^m a_{ij_\lambda} \bar{y}_i = c_{j_\lambda}, \quad \lambda = 1, 2, \dots, m, \quad (*)$$

имеющей ту же матрицу коэффициентов, что и (4.44). Здесь $A_{j_1}, \dots, A_{j_m}$ — базис $\bar{Y}$. Поскольку системы уравнений (4.44) и (*) отличаются друг от друга лишь правыми частями, их решения целесообразно находить одновременно. Если бы мы знали заранее, какие из векторов условий образуют искомый базис, поместили справа от

исходной таблицы соответствующие единичные векторы и применили схему метода градиента, то в результате на пересечении добавленных столбцов и строк, отвечающих базису, образовалась бы матрица $\|e_{st}\|$. Заметим, что преобразования метода градиента не влияют на единичный вектор, отвечающий данной строке, до тех пор, пока она не будет введена в систему равенств (4.64). Поэтому дополнение таблиц метода градиента единичными векторами можно осуществлять по мере образования искомого базиса. Это обстоятельство позволяет совместить процесс построения матрицы $\|e_{st}\|$ с процессом отыскания опорного плана Y .

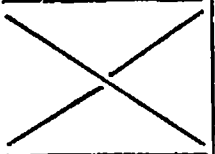
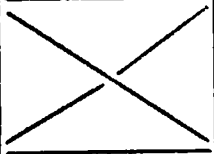
Т а б л и ц а

	1	2	3	4	5	6	7	8
0	Δ	μ	7	8	6	7	$b \diagdown c$	e_5
1	3		1	2	1	2	1	
2	6		1	3	2	1	2	
3	9		2	1	3	3	3	
4	7		3	1	2	1	1	
← 5			<u>2</u>	3	0,5	2	2	1
6	1		3	2	1	3	3	
7	3		1	2	1	1	1	
			0	0	4	0	$\diagup Y$	

В табл. 1 столбец μ остается незаполненным, поскольку в данном случае столбец Δ образуется с помощью исходного неопорного плана $Y = (0, 0, 4, 0)$: $\Delta_j = 4a_{3j} - c_j, \quad j = 1, 2, 3, 4, 5, 6, 7$. В столбце Δ имеется один нуль (позиция таблицы, в которую следует поместить нуль, остается свободной), соответствующий вектору A_5 . Поэтому табл. 1 дополняется столбцом, содержащим единичный вектор e_5 .

В 5-й строке табл. 1 выбираем произвольный (в пределах столбцов 3—6) ненулевой элемент, принадлежащий, скажем, 3-у столбцу, и, поделив на него рассматриваемую строку, переносим результат в табл. 2. Для получения i -й строки ($i \neq 5$) новой таблицы из соответствующей строки табл. 1 вычитаем 5-ю строку табл. 2, умноженную предварительно на элемент, расположенный на пересечении i -й строки и 3-го столбца старой таблицы. Описанные преобразования не касаются первых двух столбцов. В i -ю позицию столбца μ' табл. 2 записывается сумма произведений соответствующих элементов 0-й и i -й строк этой таблицы.

Т а б л и ц а 2

	1	2	3	4	5	6	7	8	9
0	Δ'	μ'		$-\frac{5}{2}$	$\frac{17}{4}$		b' c'	e_5	e_6
1	$\frac{320}{117}$	$\frac{31}{16}$		$\frac{1}{2}$	$\frac{3}{4}$	1		$-\frac{1}{2}$	
2	$\frac{643}{117}$	$\frac{59}{16}$		$\frac{3}{2}$	$\frac{7}{4}$		1	$-\frac{1}{2}$	
3	$\frac{803}{117}$	16		-2	$\frac{5}{2}$	1	1	-1	
4	$\frac{66}{13}$	$\frac{225}{16}$		$-\frac{7}{2}$	$\frac{5}{4}$	-2	-2	$-\frac{3}{2}$	
→ 5			1	$\frac{3}{2}$	$\frac{1}{4}$	1	1	$\frac{1}{2}$	
→ 6		$\frac{117}{16}$		$-\frac{5}{2}$	$\frac{1}{4}$			$-\frac{3}{2}$	1
7	$\frac{320}{117}$	$\frac{31}{16}$		$\frac{1}{2}$	$\frac{3}{4}$			$-\frac{1}{2}$	

$\theta_0' = \frac{16}{117}$

Затем составляются отношения соответствующих элементов столбцов Δ (табл. 1) и μ' (табл. 2): $\frac{\Delta_i}{\mu_i'}$ для

$\mu_i' > 0$, из которых выбирается минимальное. В нашем случае

$$\theta_0' = \min_{\mu_i' > 0} \left\{ \frac{\Delta_i}{\mu_i'} \right\} = \frac{16}{117}.$$

Для образования столбца Δ' из столбца Δ (табл. 1) вычитается столбец μ' , умноженный на θ_0' . Строка, соответствующая нулю столбца Δ' , выделяется и против нее в 9-м столбце табл. 2 помещается 1. Переход от табл. 2 к табл. 3 и так далее осуществляется по тем же правилам.

Таблица 3

	1	2	3	4	5	6	7	8	9	10
0	Δ''	μ''		40			b'' c''	e_5	e_6	e_7
1	$\frac{8}{19}$	320		8		1		4	-3	
→ 2		760		<u>19</u>			1	10	-7	1
3	$\frac{4}{19}$	920		23		1	-1	14	-10	
4	$\frac{17}{19}$	360		9		-2	-2	6	-5	
→ 5	$\times$	$\times$	1	4		1	1	2	-1	
→ 6	$\times$	$\times$		-10	1			-6	4	
7	$\frac{8}{19}$	320		8				4	-3	

$\theta_0'' = \frac{643}{117 \cdot 760}.$

При составлении табл. 4 мы сталкиваемся с особым случаем, отмеченным в п. 2.5,

$$b_1''' = b_2''' = b_3''' = b_4''' = 0.$$

В соответствии с изложенным там правилом в столбец μ''' следует перенести один из еще не окончательно пре-

Таблица 4

	1	2	3	4	5	6	7	8	9	10	11	
0	Δ'''	μ'''					c'''	e_5	e_6	e_2	e_3	
1.	$\frac{4}{19}$	1				1	$-\frac{8}{19}$	$-\frac{4}{19}$	$-\frac{1}{19}$	$-\frac{8}{19}$		
→ 2				1			$\frac{1}{19}$	$\frac{10}{19}$	$-\frac{7}{19}$	$\frac{1}{19}$		
→ 3		1				$\frac{1}{19}$	$-\frac{4}{19}$	$\frac{36}{19}$	$-\frac{29}{19}$	$-\frac{23}{19}$	1	$\theta_0''' = \frac{4}{19}$
4	$\frac{55}{19}$	-2				-2	$-\frac{47}{19}$	$\frac{24}{19}$	$-\frac{32}{19}$	$-\frac{9}{19}$		
→ 5			1			1	$\frac{15}{19}$	$-\frac{2}{19}$	$\frac{9}{19}$	$-\frac{4}{19}$		
→ 6					1		$\frac{10}{19}$	$-\frac{14}{19}$	$\frac{6}{19}$	$\frac{10}{19}$		
7	$\frac{8}{19}$	1					$-\frac{8}{19}$	$-\frac{4}{19}$	$-\frac{1}{19}$	$-\frac{8}{19}$		

образованных столбцов 3—6. В данном случае таким столбцом является лишь шестой. Он и переносится в столбец μ''' . Столбец Δ''' образуется по обычным правилам.

Заполнив табл. 4, переходим к построению окончательной табл. 5. В этой таблице столбцы μ^{IV} , Δ^{IV} остаются свободными.

Табл. 5 содержит полную информацию об искомом опорном плане сопряженной задачи $\bar{Y} = (\bar{y}_1, \bar{y}_2, \bar{y}_3, \bar{y}_4)$ и связанной с ним матрице $\|e_{st}\|$. Базис плана состоит из векторов A_2, A_3, A_5, A_6 , которые соответствуют строкам табл. 5, содержащим 1 в пределах столбцов 3—6. Для получения первой компоненты опорного плана движемся по столбцу 3 табл. 5 до 1, затем по строке, содержащей 1, до столбца 7: $\bar{y}_1 = 1$. Действуя аналогичным образом, имеем $\bar{y}_2 = \frac{1}{19}$; $\bar{y}_3 = \frac{10}{19}$; $\bar{y}_4 = -\frac{4}{19}$. Итак, $\bar{Y} = (1, \frac{1}{19}, \frac{10}{19}, -\frac{4}{19})$.

Таблица 5

	1	2	3	4	5'	6	7	8	0	10	11
0	Δ_{IV}	μ_{IV}					b_{IV} / c_{IV}	e_5	e_6	e_2	e_3
A_1	1						$-\frac{4}{19}$	$\frac{40}{-19}$	$\frac{28}{19}$	$\frac{15}{19}$	-1
A_2	2			1			$\frac{1}{19}$	$\frac{10}{19}$	$-\frac{7}{19}$	$\frac{1}{19}$	
A_3	3					1	$-\frac{4}{19}$	$\frac{36}{19}$	$-\frac{29}{19}$	$-\frac{23}{19}$	1
A_4	4						$\frac{55}{19}$	$\frac{96}{19}$	$-\frac{61}{19}$	$-\frac{55}{19}$	2
A_5	5		1				1	-2	2	1	-1
A_6	6				1		$\frac{10}{19}$	$-\frac{14}{19}$	$\frac{6}{19}$	$\frac{10}{19}$	
A_7	7						$-\frac{8}{19}$	$-\frac{4}{19}$	$-\frac{1}{19}$	$-\frac{8}{19}$	

Переходим к заполнению первой основной табл. Ia метода уточнения оценок.

Т а б л и ц а I a

$B_1 \backslash e$	e_0	e_1	e_2	e_3	e_4	A_i
A_2	$\frac{40}{19}$	1	$\frac{1}{19}$	$\frac{10}{19}$	$-\frac{23}{19}$	$\frac{55}{19}$
A_3		-1			1	-2
A_5	$-\frac{18}{19}$	-2	$\frac{10}{19}$	$-\frac{14}{19}$	$\frac{36}{19}$	$-\frac{96}{19}$
A_6	$\frac{43}{19}$	2	$-\frac{7}{19}$	$\frac{6}{19}$	$-\frac{29}{19}$	$\frac{90}{19}$
	$\frac{173}{19}$	1	$\frac{1}{19}$	$\frac{10}{19}$	$-\frac{4}{19}$	$\frac{55}{19}$

Компоненты $\bar{Y}$ вписываем в последнюю строку табл. Ia (столбцы e_1, e_2, e_3, e_4). Строка A_i табл. Ia образуется точно так же, как и последняя, с той разницей, что ее элементы извлекаются из столбца e_i табл. 5. В столбец e_0 помещаются коэффициенты разложения вектора B по базису B_1 (A_2, A_3, A_5, A_6), вычисляемые по формуле

$$e_{i0} = \sum_{t=1}^4 b_t e_{it}.$$

Поскольку среди элементов столбца e_0 имеются отрицательные, необходимо, заполнив последний столбец табл. Ia, перейти к построению табл. Ib. Предварительно выпишем вспомогательную табл. II. Первые 5 строк табл. II заполняются, как обычно, компонентами векторов условий $A_1, A_2, A_3, A_4, A_5, A_6, A_7$ и коэффициентами линейной формы задачи. В строку Δ поместим величины

$$\Delta_j = \sum_{i=1}^4 a_{ij} \bar{y}_i - c_j, \quad j = 1, 2, 3, 4, 5, 6, 7,$$

где $\bar{Y} = (\bar{y}_1, \bar{y}_2, \bar{y}_3, \bar{y}_4)$ — определенный только что опорный план сопряженной задачи.

Таблица II

$N \backslash c$	1	2	3	1	2	3	1	B	A_5
1	1	1	2	3	2	3	1	7	-2
2	2	3	1	1	3	2	2	8	$\frac{10}{19}$
3	1	2	3	2	$\frac{1}{2}$	1	1	7	$-\frac{14}{19}$
4	2	1	3	1	2	3	1	6	$\frac{36}{19}$
Δ	$\frac{4}{19}$			$\frac{55}{19}$			$\frac{8}{19}$	$\times$	$\times$
$z_{(5)}$	$\frac{40}{19}$			$-\frac{96}{19}$	1		$\frac{4}{19}$	$\times$	$\times$
$-\frac{\Delta}{z_{(5)}}$				$\frac{55}{96}$				$\times$	$\times$
Δ'	$\frac{17}{12}$				$\frac{55}{96}$		$\frac{13}{24}$	$\times$	$\times$

$$\theta_0 = \frac{55}{96}.$$

Единственный отрицательный элемент столбца e_0 табл. Ia расположен в строке A_5 . Следовательно, вектор A_5 подлежит выводу из базиса. Элементы строки A_5 : $e_{51}, e_{52}, e_{53}, e_{54}$ записываем в столбец A_5 табл. II. Затем, используя формулу

$$z_{5j} = \sum_{t=1}^4 e_{5t} a_{tj}, \quad j = 1, 4, 7,$$

заполняем строку $z_{(5)}$ табл. II (остальные элементы этой строки определяются автоматически $z_2 := z_3 = z_6 = 0; z_5 = 1$). Далее вычисляем $-\frac{\Delta_j}{z_{5j}}$ для $z_{5j} < 0$ и результат помещаем в строку $-\frac{\Delta}{z_{(5)}}$. В нашем случае строка $-\frac{\Delta}{z_{(5)}}$ содержит только один элемент (лишь $z_{54} = -\frac{96}{19} < 0$).

Поэтому

$$\theta_0 = \min_{z_{5j} < 0} \left(-\frac{\Delta_j}{z_{5j}} \right) = \frac{\Delta_4}{z_{54}} = \frac{55}{96}$$

и в базис B_2 следует включить вектор A_4 . Определяем, пользуясь формулами

$$z_{i4} = \sum_{t=1}^4 e_{it} a_{t4}, \quad i = 2, 3, 5, 6,$$

коэффициенты разложения вектора A_4 по базису B_1 и располагаем их в столбец A_4 табл. 1а. В правый нижний угол табл. 1а помещаем $\Delta_4 = \frac{55}{19}$, в левый нижний угол

$$\sum_{i=1}^4 b_i \bar{y}_i = \frac{173}{19}.$$

Теперь мы можем начать построение табл. 1б.

Т а б л и ц а 1б

$B_2 \backslash l$	e_0	e_1	e_2	e_3	e_4	
A_2	$\frac{25}{16}$	$-\frac{7}{48}$	$\frac{17}{48}$	$\frac{5}{48}$	$-\frac{1}{8}$	
A_3	$\frac{6}{16}$	$-\frac{5}{24}$	$-\frac{5}{24}$	$\frac{7}{24}$	$\frac{1}{4}$	
$\rightarrow A_4$	$\frac{3}{16}$	$\frac{19}{48}$	$-\frac{5}{48}$	$\frac{7}{48}$	$-\frac{3}{8}$	
A_6	$\frac{11}{8}$	$\frac{1}{8}$	$\frac{1}{8}$	$-\frac{3}{8}$	$\frac{1}{4}$	
	$\frac{137}{16}$	$-\frac{7}{48}$	$\frac{17}{48}$	$\frac{5}{48}$	$\frac{7}{8}$	

Строка A_4 табл. 1б образуется из строки A_5 табл. 1а делением элементов последней на величину, расположенную на пересечении строки A_5 и столбца A_4 . Вычитая далее из остальных строк табл. 1а строку A_4 табл. 1б, умноженную на последний элемент уменьшаемой строки, получаем соответствующие строки табл. 1б. Все величины столбца e_0 новой основной таблицы неотрицатель-

ны. Следовательно, исследуемая задача линейного программирования уже решена. Оптимальный план исходной задачи X^* обладает базисом $B_2: A_2, A_3, A_4, A_6$, его ненулевые компоненты расположены в столбце c_0 табл. 16

$$X^* = \left(0, \frac{25}{16}, \frac{3}{8}, \frac{3}{16}, 0, \frac{11}{8}, 0 \right).$$

Решение сопряженной задачи извлекается из последней строки табл. 16.

$$Y^* = \left(-\frac{7}{48}, \frac{17}{48}, \frac{5}{48}, \frac{7}{48} \right).$$

Рекомендуем читателю проверить, что векторы X^* , Y^* действительно удовлетворяют условиям исходной и сопряженной задач соответственно, причем

$$\sum_{j=1}^7 c_j x_j^* = \sum_{i=1}^4 b_i y_i^* = \frac{137}{16}.$$

Для решения задачи в данном случае потребовалось всего 1 итерация метода уточнения оценок. Это объясняется тем, что в результате использования метода градиента было получено хорошее первое приближение. Следует заметить, что такое положение не случайно: метод градиента приводит обычно к опорному плану, близкому к оптимальному.

§ 3. МЕТОД ПОСЛЕДОВАТЕЛЬНОГО СОКРАЩЕНИЯ НЕВЯЗОК

3.1. Предварительные замечания. В частящем параграфе будет рассмотрен еще один общий метод решения задач линейного программирования, существенно опирающийся на идеи двойственности. Метод последовательного улучшения плана оперирует лишь с исходной задачей. Метод последовательного уточнения оценок имеет дело только с сопряженной задачей. Основная особенность излагаемого ниже метода последовательного сокращения невязок состоит в совместном использовании обеих задач двойственной пары. Идея этого метода была высказана еще в 1939 г. Л. В. Канторовичем, применившим ее к ряду частных задач линейного программирования [6], [12] *).

*) Л. В. Канторович, не рассматривая сопряженной задачи, ввел понятие разрешающих множителей, совокупность которых, как мы видели в § 1 настоящей главы, составляет ее оптимальный план.

Независимое изложение метода сокращения невязок было дано в 1956 г. тремя американскими авторами [3], перенесшими так называемый венгерский метод (гл. 5, § 3) на случай общей задачи линейного программирования. В методе сокращения невязок, так же как и в методе уточнения оценок, достижение оптимального плана исследуемой задачи осуществляется путем движения по векторам X , каждый из которых не является ее планом. Однако в отличие от метода уточнения оценок в рассматриваемом методе все промежуточные вектора обладают неотрицательными компонентами [они не обязаны удовлетворять системе условий (4.28)].

Итак, пусть необходимо решить задачу линейного программирования, записанную в канонической форме V. Без потери общности можно считать все компоненты вектора ограничений $B = (b_1, b_2, \dots, b_m)$ неотрицательными. Действительно, если бы для некоторого b_i это условие не выполнялось, мы перешли бы к эквивалентной задаче путем умножения i -го соотношения системы (4.28) на -1 .

Введем в рассмотрение задачу о минимизации линейной формы

$$\sum_{i=1}^m \varepsilon_i \quad (4.87)$$

при соблюдении ограничений

$$\sum_{j=1}^n a_{ij}x_j + \varepsilon_i = b_i; \quad i = 1, 2, \dots, m; \quad (4.88)$$

$$x_j \geq 0; \quad j = 1, 2, \dots, n; \quad \varepsilon_i \geq 0; \\ i = 1, 2, \dots, m. \quad (4.89)$$

Эту задачу будем в дальнейшем называть расширенной. Общая схема метода последовательного сокращения невязок состоит в следующем. Прежде всего выберем некоторый план сопряженной задачи VI $Y = (y_1, y_2, \dots, y_m)$ (не обязательно опорный) и отметим те из векторов условий $A_j, j = 1, 2, \dots, n$, для которых

$$\sum_{i=1}^m a_{ij}y_i = c_j.$$

(Способ отыскания исходного плана сопряженной задачи был рассмотрен в п. 2.5). Множество индексов таких векторов обозначим через E_Y . С планом $Y = (y_1, y_2, \dots, y_m)$ свяжем вспомогательную линейную задачу, образуемую из расширенной [(4.87), (4.88), (4.89)] путем введения дополнительных ограничений

$$x_j = 0 \text{ при } j \in E_Y.$$

Таким образом, векторами условий вспомогательной задачи являются $A_{j_1}, A_{j_2}, \dots, A_{j_s}; e_1, e_2, \dots, e_m$, где индексы $j_1, j_2, \dots, j_s$ составляют множество E_Y , $e_1, e_2, \dots, e_m$ — полный набор единичных m -мерных векторов. Обычно $s \leq m \ll n$. Поэтому исследование вспомогательной задачи проводится сравнительно просто. Решив вспомогательную задачу, определяем оптимальный план задачи, с ней сопряженной, используя который, строим далее новый план задачи VI $\bar{Y} = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_m)$. После этого описанная последовательность действий (составляющая отдельную итерацию метода) повторяется, отправляясь от плана $\bar{Y} = (\bar{y}_1, \bar{y}_2, \dots, \bar{y}_m)$. При этом минимальное значение линейной формы новой вспомогательной задачи оказывается уменьшенным по сравнению с минимумом предшествующей вспомогательной задачи. Процесс продолжается до тех пор, пока линейная форма очередной вспомогательной задачи не станет равной нулю.

Пусть $\sigma_1, \sigma_2, \dots, \sigma_s, e_1, e_2, \dots, e_m$ — произвольный план некоторой вспомогательной задачи, связанной, например, с планом Y . Каждому такому плану соответствует n -мерный вектор $X = (x_1, x_2, \dots, x_n)$, где

$$\begin{aligned} x_\lambda &= \sigma_\lambda; \quad \lambda = 1, 2, \dots, s; \\ x_j &= 0, \text{ если } j \in E_Y. \end{aligned} \tag{4.90}$$

Вектор X , вообще говоря, не является планом исходной задачи V: в соответствии с условиями (4.88)

$$b_i - \sum_{j=1}^n a_{ij} x_j = e_i.$$

Таким образом, величина ε_i определяет невязку, возникающую в i -м уравнении системы (4.28) при подстановке в нее вектора $X = (x_1, \dots, x_n)$. Рассматриваемый метод ведет к последовательному сокращению суммы этих невязок (отсюда и его название). Через конечное число итераций сумма невязок становится равной нулю. Тогда вектор $X^* = (x_1^*, x_2^*, \dots, x_n^*)$, связанный с оптимальным планом соответствующей вспомогательной задачи соотношениями (4.90), являясь, очевидно, планом задачи V , совпадает вместе с тем с ее решением.

3.2. Описание метода. Переходим к детальному рассмотрению отдельных элементов метода. Допустим, что решения вспомогательной задачи, связанной с $Y = (y_1, y_2, \dots, y_m)$, и задачи, с ней сопряженной, получены (это делается обычно с помощью метода улучшения плана). Обозначим их соответственно через

$$(\Sigma^*; E^*) = (\sigma_1^*, \sigma_2^*, \dots, \sigma_s^*; \varepsilon_1^*, \varepsilon_2^*, \dots, \varepsilon_m^*) \text{ и} \\ M^* = (\mu_1^*, \mu_2^*, \dots, \mu_m^*).$$

Отметим некоторые свойства векторов $(\Sigma^*; E^*)$; M^* . Для этого прежде всего сформулируем задачу, сопряженную вспомогательной (в дальнейшем она называется сопряженной вспомогательной задачей).

Обратить в максимум линейную форму

$$\sum_{i=1}^m b_i \mu_i, \quad (4.91)$$

переменные которой удовлетворяют условиям

$$\sum_{i=1}^m a_{ij} \mu_i \leq 0 \text{ при } j \in E_Y; \quad (4.92)$$

$$\mu_i \leq 1; \quad i = 1, 2, \dots, m. \quad (4.93)$$

Таким образом, вектор $M^* = (\mu_1^*, \mu_2^*, \dots, \mu_m^*)$ удовлетворяет системам неравенств (4.92), (4.93). Кроме того, в силу второй теоремы двойственности при

$$\sigma_\lambda^* > 0 \quad \sum_{i=1}^m a_{ij\lambda} \mu_i^* = 0. \quad (4.94)$$

Решение вспомогательной задачи может привести к одной из следующих альтернатив:

$$\text{I. } \sum_{i=1}^m \varepsilon_i^* = 0;$$

$$\text{II. } \sum_{i=1}^m \varepsilon_i^* > 0.$$

В первом случае, как мы сейчас убедимся, оптимальным планом задачи V является вектор $X = (x_1, x_2, \dots, x_n)$, координаты которого связаны с $\Sigma^* = (\sigma_1^*, \sigma_2^*, \dots, \sigma_s^*)$ соотношениями (4.90).

В самом деле, все компоненты вектора X неотрицательны. Далее из неотрицательности каждого из

ε_i^* и равенства $\sum_{i=1}^m \varepsilon_i^* = 0$ получаем: $\varepsilon_i^* = 0, \quad i = 1,$

$2, \dots, m$. Поэтому вектор X , удовлетворяя условиям (4.28) задачи V, является ее планом. Согласно определению плана $X = (x_1, x_2, \dots, x_m)$ при $x_j > 0$

$\sum_{i=1}^m a_{ij}y_i = c_j$. Следовательно,

$$\sum_{j=1}^n c_j x_j = \sum_{j=1}^n x_j \sum_{i=1}^m a_{ij} y_i = \sum_{i=1}^m y_i \sum_{j=1}^n a_{ij} x_j = \sum_{i=1}^m b_i y_i.$$

Полученное равенство в силу следствия 2 первой теоремы двойственности указывает на оптимальность плана X (равно, как и на оптимальность плана Y задачи VI).

Таким образом, если исследование вспомогательной задачи приводит к альтернативе I, ее решение определяет оптимальный план исходной задачи и весь процесс на этом заканчивается. Переходим к анализу альтернативы II.

Положим,

$$\delta_j = \sum_{i=1}^m a_{ij} y_i - c_j, \quad j = 1, 2, \dots, n; \quad (4.95)$$

$$\Delta_j^* = \sum_{i=1}^m a_{ij} \mu_i^*, \quad j = 1, 2, \dots, n. \quad (4.96)$$

Пусть имеет место альтернатива II:

$$\sum_{i=1}^m \epsilon_i^* > 0.$$

В таком случае необходимо рассмотреть две возможности:

1°. Среди величин Δ_j^* , определяемых по формуле (4.96), найдется хотя бы одна положительная.

2°. $\Delta_j^* \leq 0$ для $j = 1, 2, \dots, n$.

В первом случае (1°) строится новый план задачи VI $\bar{Y} = (\bar{y}_1, \dots, \bar{y}_m)$, вспомогательная задача для которого обладает меньшим минимумом по сравнению с предыдущей.

Положим,

$$\bar{y}_i(\theta) = y_i - \theta \mu_i^*, \quad i = 1, 2, \dots, m.$$

В соответствии с обозначениями (4.95), (4.96)

$$\bar{\delta}_j(\theta) = \sum_{i=1}^m a_{ij} \bar{y}_i(\theta) - c_j = \delta_j - \theta \Delta_j^*, \quad (4.97)$$

$$j = 1, 2, \dots, n.$$

Определим максимально возможное значение θ , при котором вектор $\bar{Y}(\theta)$ все еще остается планом задачи VI. Согласно (4.89) искомое значение θ определяется формулой

$$\theta_0 = \min_{\Delta_j^* > 0} \left(\frac{\delta_j}{\Delta_j^*} \right), \quad (4.98)$$

где минимум берется по тем индексам j , для которых $\Delta_j^* > 0$ (поскольку мы имеем дело со случаем 1°, такие индексы обязательно найдутся). Пользуясь определением величин Δ_j^* и неравенствами (4.92), получаем

$$\Delta_j^* \leq 0 \quad \text{при } j \in E_Y.$$

С другой стороны, по определению множества индексов E_Y $\delta_j > 0$ при $j \in E_Y$. Следовательно, $\theta_0 > 0$. Искомый новый план $\bar{Y}$ задачи VI определяется соотношением

$$\bar{Y} = \bar{Y}(\theta_0) = Y - \theta_0 M^*. \quad (4.99)$$

Проверим, что он приводит к уменьшению линейной формы задачи VI. Действительно,

$$\sum_{i=1}^m b_i \bar{y}_i = \sum_{i=1}^m b_i y_i - \theta_0 \sum_{i=1}^m b_i \mu_i^*.$$

Согласно первой теоремы двойственности

$$\sum_{i=1}^m b_i \mu_i^* = \sum_{i=1}^m \varepsilon_i^*$$

[теорема применяется к вспомогательной задаче и сопряженной вспомогательной задаче (4.91), (4.92), (4.93)]. Наше утверждение, таким образом, вытекает из положительности θ_0 и $\varepsilon_0 = \sum_{i=1}^m \varepsilon_i^*$. После образования нового плана сопряженной задачи VI для него составляется вспомогательная задача, и вся последовательность действий повторяется.

Проследим за изменениями, происходящими в процессе решения с суммой невязок. Прежде всего заметим, что согласно условию (4.94) в $E_{\bar{y}}$ входят, в частности, те индексы j , для которых $x_j = \sigma_j^* > 0$. Действительно, для таких j

$$\delta_j = \delta_j - \theta_0 \Delta_j^* = 0 - \theta_0 \cdot 0 = 0.$$

Поэтому исходный план новой вспомогательной задачи можно составить из положительных компонент оптимального плана предыдущей (это обстоятельство, кстати, существенно сокращает число шагов, необходимых для решения очередной вспомогательной задачи). Определим индекс j_0 соотношением

$$\theta_0 = \frac{\delta_{j_0}}{\Delta_{j_0}^*} \quad (\Delta_{j_0}^* > 0).$$

Таких индексов может быть несколько.

Очевидно, вектор A_{j_0} войдет в число векторов условий новой вспомогательной задачи ($\bar{\delta}_{j_0} = \delta_{j_0} - \theta_0 \Delta_{j_0}^* = 0$).

Допустим, что эта задача невырожденная [для этого достаточно предположить невырожденность расширенной задачи (4.87), (4.88), (4.89)]. В таком случае

условие $\Delta_{j_0}^* > 0$ означает, что введение в базис исходного плана новой вспомогательной задачи вектора A_{j_0} уменьшит значение ее линейной формы (см. описание II алгоритма метода последовательного улучшения плана, гл. 3, § 6). Поэтому минимум новой вспомогательной задачи строго меньше минимума предыдущей вспомогательной задачи

$$\sum_{i=1}^m \bar{\epsilon}_i^* < \sum_{i=1}^m \epsilon_i^*. \quad (4.100)$$

Следует заметить, что при вырожденности расширенной задачи неравенство (4.100) может перейти в равенство.

Обратимся теперь к случаю 2°. Пусть некоторая итерация привела к величинам $\Delta_j^* \leq 0$, $j = 1, 2, \dots, n$, причем $\epsilon_0 = \sum_{i=1}^m \epsilon_i^* > 0$. Тогда вектор $\bar{Y}(\theta) = Y - \theta M^*$ является планом задачи VI при любом $\theta \geq 0$ [соотношение (4.97)].

С другой стороны,

$$\sum_{i=1}^m b_i \bar{y}_i(\theta) = \sum_{i=1}^m b_i y_i - \theta \sum_{i=1}^m b_i \mu_i^* = \sum_{i=1}^m b_i y_i - \theta \sum_{i=1}^m \epsilon_i^*.$$

Полученное равенство означает, что линейная форма задачи VI не ограничена снизу на множестве ее планов. Следовательно, исходная задача не имеет ни одного плана (первая теорема двойственности). Итак, наличие случая 2° указывает на неразрешимость исследуемой задачи.

Резюмируем кратко полученные выводы. Каждая итерация метода может завершиться либо альтернативой I, либо альтернативой II. При наличии альтернативы I решение последней вспомогательной задачи определяет оптимальный план задачи V. Если же имеет место альтернатива II, следует рассмотреть два случая 1°, 2°.

В первом случае указывается способ образования нового плана сопряженной задачи VI, отправляясь от которого, проводится следующая итерация. Что касается случая 2°, то его наличие указывает на неразрешимость исходной задачи V.

Каждая итерация метода сокращения невязок включает несколько шагов метода улучшения плана, необходимых для решения соответствующей вспомогательной задачи. Заметим, что планы любой вспомогательной задачи являются планами расширенной задачи (4.87), (4.88), (4.89). Таким образом, процесс решения произвольной вспомогательной задачи состоит в последовательных переходах от одного опорного плана расширенной задачи к другому, осуществляемых по правилам метода улучшения плана. Однако было бы неправильным считать, что описываемый метод решения исходной задачи V эквивалентен исследованию расширенной задачи методом последовательного улучшения плана. В самом деле, на протяжении отдельной итерации метода сокращения невязок вводимый вектор выбирается не из всей совокупности векторов условий расширенной задачи, а лишь из числа единичных векторов $e_1, e_2, \dots, e_m$ и тех A_j , для которых

$$\sum_{i=1}^m a_{ij} y_i = c_j$$

(здесь под вектором $Y = (y_1, y_2, \dots, y_m)$ понимается план сопряженной задачи VI , связанный с данной итерацией). Перед началом каждой итерации множество допустимых (в предшествующей итерации) векторов условий обновляется с помощью нового плана задачи VI .

Приведенные соображения позволяют, в частности, сделать вывод о конечности исследуемого метода. Действительно, если расширенная задача невырожденная, движение от одного ее опорного плана к другому сопровождается монотонным убыванием линейной формы (4.87), что гарантирует их различие. Но количество различных опорных планов расширенной задачи конечно. Следовательно, при условии разрешимости задачи V (случай 2° отсутствует) получение ее оптимального плана методом сокращения невязок укладывается в конечное число итераций, каждая из которых составляется из конечного же числа шагов. Если при выборе вектора, подлежащего выводу из базиса очередной вспомогательной задачи, пользоваться правилами, изложенными в § 7 гл. 3, то это утверждение будет иметь место и в случае вырожденности расширенной задачи.

Скажем пару слов о геометрии метода сокращения невязок в $(m + 1)$ -мерном пространстве. Решение задачи V начинается с выбора некоторого плана задачи Y VI. Геометрическим аналогом Y является гиперплоскость Π , проходящая через начало координат и расположенная «над» конусом задачи V K (п. 1.6 настоящей главы). Затем фиксируется общая часть Π и K , которая также является многогранным конусом. Обозначим этот конус через K_Π . Размерность конуса K_Π не превышает m (если Y — опорный план задачи VI, размерность K_Π в точности равна m). В случае $m=2$ конус K_Π может совпадать либо с точкой $(0, 0, 0)$, либо с лучом, исходящим из начала координат, либо с двумерным конусом.

Из конуса K_Π с помощью условий

$$z_i \leq b_i, \quad i = 1, 2, \dots, m$$

высекается многогранная область $(K_\Pi)_B$. Среди точек этой области выбирается наиболее близкая к крайней «верхней» точке пересечения K и $\mathcal{L}$ (точка Z) *). Если точка Z совпадает с крайней точкой пересечения K и $\mathcal{L}$ (Z^*), процесс решения заканчивается (оптимальный план исследуемой задачи определяется компонентами разложения $Z = Z^*$ по векторам, порождающим конус K_Π). В противном случае осуществляется поворот гиперплоскости, приводящий к новому конусу K_{Π_1} . При этом найденная на предыдущей итерации точка Z оказывается принадлежащей K_{Π_1} . Затем, отправляясь от Z , производится поиск вектора Z_1 , принадлежащего $(K_{\Pi_1})_B$ и наименее уклоняющегося от Z^* . Процесс решения завершается получением при некотором N вектора $Z_N = Z^*$.

Таким образом, при использовании метода последовательного сокращения невязок движение к крайней верхней точке пересечения K и $\mathcal{L}$, соответствующей оптимальному плану исследуемой задачи, осуществляется по границе конуса K . В этой связи полезно вспомнить, что метод улучшения плана ведет к искомой крайней точке изнутри K , метод уточнения оценок — извне K .

*) Здесь расстояние между точками $Z_1 = (z_1^{(1)}, z_2^{(1)}, \dots, z_{m+1}^{(1)})$, $Z_2 = (z_1^{(2)}, z_2^{(2)}, \dots, z_{m+1}^{(2)})$ измеряется величиной $\sum_{i=1}^m |z_i^{(1)} - z_i^{(2)}|$.

3.3. Алгоритм метода. Как уже отмечалось, отдельная итерация метода последовательного сокращения невязок состоит из нескольких шагов метода улучшения плана, примененного к соответствующей вспомогательной задаче. Выбор для исследования вспомогательных задач метода улучшения плана обусловлен тем, что исходный план первой из них строится автоматически (его базис состоит из единичных векторов $e_1, e_2, \dots, e_m$), а в качестве исходного плана каждой последующей вспомогательной задачи можно принять решение предыдущей. Для перехода к следующей итерации нам необходимо иметь не только решение вспомогательной задачи, но и оптимальный план сопряженной вспомогательной задачи. Поэтому при решении вспомогательных задач метод улучшения плана разумно использовать в форме II алгоритма *). Эти предварительные замечания позволяют при реализации метода использовать незначительно измененную схему II алгоритма метода улучшения плана (§ 6 гл. 3).

Алгоритм метода последовательного сокращения невязок состоит в заполнении одной вспомогательной таблицы и ряда основных таблиц. Для удобства нумерация таблиц и обозначения, принятые в § 6 гл. 3, сохраняются. Поскольку единичные векторы $e_1, e_2, \dots, e_m$ входят в систему векторов условий любой вспомогательной задачи, заполнение главной части первой основной таблицы (табл. Ia) не составляет труда

$$e_{ij} = \begin{cases} 0, & \text{если } i \neq j, \\ 1, & \text{если } i = j, \end{cases} \quad i, j = 1, 2, \dots, m;$$

$$e_{i0} = b_i, \quad i = 1, 2, \dots, m;$$

$$e_{m+1,0} = \sum_{i=1}^m e_{i0} = \sum_{i=1}^m b_i,$$

$$e_{m+1,j} = \mu_j = \sum_{i=1}^m c_i e_{ij} = 1, \quad j = 1, 2, \dots, m,$$

(в нашем случае $c_1 = c_2 = \dots = c_m = 1$).

*) Как нетрудно проверить (предоставляем это читателю), при использовании II алгоритма нижняя строка последней основной таблицы заполняется компонентами оптимального плана сопряженной задачи (§ 6 гл. 3).

Таким образом, табл. Ia имеет следующий вид.

Т а б л и ц а I a

c	$B_1 \backslash e$	e_0	e_1	e_2	$\dots$	e_k	$\dots$	e_m	A_1
1	e_1	b_1	1		$\dots$		$\dots$		a_{11}
1	e_2	b_2		1	$\dots$		$\dots$		a_{21}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
1	e_k	b_k			$\dots$		$\dots$		a_{k1}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
1	e_m	b_m			$\dots$		$\dots$	1	a_{m1}
		$e_{m+1,0}$	1	1	$\dots$	1	$\dots$	1	Δ_1

Как обычно, клетки таблицы, в которых следует поместить нуль, остаются пустыми. Для получения элементов последнего столбца табл. Ia необходимо обратиться к вспомогательной табл. II, заполнение которой продолжается в течение всего процесса решения задачи. При анализе вспомогательных задач мы будем придерживаться следующего правила: произвольный единичный вектор e_s , вышедший на некотором шаге из базиса, в последующие базисы включению не подлежит. Заметим, что иногда это ограничение может привести к завышению минимума отдельной вспомогательной задачи. Однако оно дает возможность не исследовать в процессе решения единичные векторы $e_1, e_2, \dots, e_m$ и, следовательно, не включать их в табл. II *). Заполнение табл. II начинается, как обычно, с записи компонент векторов условий $A_1, A_2, \dots, A_n$ и коэффициентов линейной формы (4.27) $c_1, c_2, \dots, c_n$.

*) Нетрудно проверить (предоставляем это читателю), что данное ограничение в выборе вводимых векторов не сказывается на сходимости изучаемого метода.

c	c _i	c ₂	...	c _j	c _n	I				II				
						μ	μ'	...	μ*	μ'	μ''	...	μ*	
A	A ₁	A ₂	...	A _j	A _n									
1	a ₁₁	a ₁₂	...	a _{1j}	a _{1n}	μ ₁	μ' ₁	...	μ* ₁	μ' ₁	μ'' ₁	...	μ* ₁	...
2	a ₂₁	a ₂₂	...	a _{2j}	a _{2n}	μ ₂	μ' ₂	...	μ* ₂	μ' ₂	μ'' ₂	...	μ* ₂	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
i	a _{i1}	a _{i2}	...	a _{ij}	a _{in}	μ _i	μ' _i	...	μ* _i	μ' _i	μ'' _i	...	μ* _i	...
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
m	a _{m1}	a _{m2}	...	a _{mj}	a _{mn}	μ _m	μ' _m	...	μ* _m	μ' _m	μ'' _m	...	μ* _m	...
δ	×	δ ₂	...	δ _j	×	I								
Δ	Δ ₁	×	...	×	Δ _n									
Δ'	Δ' ₁	×	...	×	Δ' _n									
...	...	...	...	...	...									
Δ*	Δ* ₁	Δ* ₂	...	Δ* _j	Δ* _n									
δ̄	×	δ̄ ₂	...	×	×	II								
Δ̄	Δ̄ ₁	×	...	Δ̄ _j	Δ̄ _n									
Δ̄'	Δ̄' ₁	×	...	Δ̄' _j	Δ̄' _n									
...	...	...	...	...	...									
Δ̄*	Δ̄* ₁	Δ̄* ₂	...	Δ̄* _j	Δ̄* _n									
...	...	...	...	...	...									

Пусть $Y = (y_1, y_2, \dots, y_m)$ — начальный план задачи VI. Используя этот вектор, определяем вели-

чины $\delta_j = \sum_{i=1}^m a_{ij}y_i - c_j, \quad j = 1, 2, \dots, m$ и помещаем

их в строку δ табл. II. На протяжении первой итерации исследованию подвергаются лишь те из векторов условий A_j , для которых $\delta_j = 0$. Для выделения этих векторов (совокупность их индексов обозначалась ранее через E_Y) в строку δ вместо нулей ставятся крестики ($\times$). Например, в строке δ табл. II крестиками отмечены векторы A_1, A_n . Таким образом, здесь $E_Y = (1, n)$. Далее к части табл. II, соответствующей в нашем случае векторам A_1, A_n , применяется II алгоритм метода последовательного улучшения плана, подробное описание которого дано в § 6 гл. 3. Очевидно, на протяжении первой итерации заполняются лишь те позиции строк $\Delta, \Delta', \Delta'', \dots$, которые соответствуют векторам A_1, A_n . Поскольку коэффициенты линейной формы вспомогательной задачи, относящиеся к любому из векторов A_j , равны нулю, элементы строк $\Delta, \Delta', \Delta'', \dots, \Delta^{(l)}, \dots$ вычисляются по формулам

$$\Delta_j^{(l)} = \sum_{i=1}^m a_{ij}\mu_i^{(l)} \quad \text{для } j \in E_Y. \quad (4.101)$$

Для удобства подсчета $\Delta_j^{(l)}$ величины $\mu_i^{(l)} = e_{m+1,i}^{(l)}, i = 1, 2, \dots, m$ имеет смысл записывать справа от табл. II в виде отдельного столбца.

После заполнения строки Δ выбирается ее максимальный положительный*) элемент, который определяет вектор, подлежащий вводу в новый базис. Положим, например, что таким вектором является A_1 . Тогда в последний столбец табл. Ia помещаются коэффициенты разложения вектора A_1 по векторам базиса этой таблицы $(e_1, e_2, \dots, e_m)$, совпадающие, очевидно, в данном случае с компонентами A_1 . Напомним, что в общем

*) Выбор максимального положительного (а не минимального отрицательного) элемента обусловлен тем, что решается задача минимизации.

случае элементы последнего столбца основной таблицы вычисляются по формулам

$$z_{ik} = \sum_{s=1}^m e_{is}^{(i)} a_{sk}, \quad i = 1, 2, \dots, m. \tag{4.102}$$

Здесь k — номер вводимого в базис вектора. После окончательного заполнения первой основной таблицы переходим к построению табл. Iб и следующей строки вспомогательной табл. II (строка Δ'). Все элементы табл. Iб (кроме принадлежащих последнему столбцу) вычисляют, исходя из соответствующих элементов табл. Ia с помощью выведенных в § 6 гл. 3 рекуррентных формул (3.35), (3.36), (3.37). Последний столбец этой таблицы заполняется после образования строки Δ' табл. II. Мы не будем здесь подробно описывать переход от одной основной таблицы к другой внутри отдельной итерации, а отошлем читателя к § 6 гл. 3, в котором этот переход излагается детально.

Т а б л и ц а Iб

c	$\begin{matrix} e \\ B_2 \end{matrix}$	e_0	e_1	e_2	$\dots$	e_i	$\dots$	e_m	A_n
1	e_1	e'_{10}	e'_{11}	e'_{12}	$\dots$	e'_{1i}	$\dots$	e'_{1m}	z'_{1n}
1	e_2	e'_{20}	e'_{21}	e'_{22}	$\dots$	e'_{2i}	$\dots$	e'_{2m}	z'_{2n}
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
1	e_{r-1}	$e'_{r-1,0}$	$e'_{r-1,1}$	$e'_{r-1,2}$	$\dots$	$e'_{r-1,i}$	$\dots$	$e'_{r-1,m}$	$z'_{r-1,n}$
	A_1	$e'_{r,0}$	$e'_{r,1}$	$e'_{r,2}$	$\dots$	$e'_{r,i}$	$\dots$	e'_{rm}	z'_{rn}
1	e_{r+1}	$e'_{r+1,0}$	$e'_{r+1,1}$	$e'_{r+1,2}$	$\dots$	$e'_{r+1,i}$	$\dots$	$e'_{r+1,m}$	$z'_{r+1,n}$
$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$	$\dots$
1	e_m	e'_{m0}	e'_{m1}	e'_{m2}	$\dots$	e'_{mi}	$\dots$	e'_{mm}	z'_{mn}
		$e'_{m+1,0}$	$e'_{m+1,1}$	$e'_{m+1,2}$	$\dots$	$e'_{m+1,i}$	$\dots$	$e'_{m+1,m}$	Δ'_n

При составлении табл. Iб предполагалось, что для включения в следующий базис выбран вектор A_n . Заполнив несколько основных таблиц, мы обязательно получим строку вспомогательной табл. II, состоящую из неположительных элементов (имеются в виду лишь те позиции этой строки, которые соответствуют векторам условий первой вспомогательной задачи).

В табл. II она обозначена через Δ^* . Если элемент $e_{m+1,0}^*$ расположенный в левом нижнем углу последней основной таблицы, равен нулю (или, что то же самое, базис этой таблицы не содержит векторов e_i , $i = 1, 2, \dots, m$), процесс решения заканчивается, причем положительные компоненты искомого оптимального плана содержатся в столбце e_0 рассматриваемой таблицы (альтернатива I).

В противном случае (альтернатива II) в соответствии с формулой (4.96) осуществляется заполнение свободных позиций строки Δ^* . Если все вновь определенные элементы окажутся неположительными (случай 2⁰), процесс заканчивается установлением неразрешимости исследуемой задачи. Если же среди элементов строки Δ^* имеются положительные (случай 1⁰), переходим к следующей итерации. Для этого составляются отношения $\frac{\delta_s}{\Delta_s^*}$ для $\Delta_s^* > 0$, из которых выбирается мини-

мальное. Обозначим эту величину через θ_0 . Строка $\bar{\delta}$, отправляясь от которой, проводим итерацию 2, образуется путем умножения элементов строки Δ^* на θ_0 и последующего вычитания результата из строки δ

$$\bar{\delta}_j = \delta_j - \theta_0 \Delta_j^*, \quad i = 1, 2, \dots, n.$$

Нулевые элементы вновь полученной строки помечаются крестиками. Тем самым выделяется новая система векторов условий, соответствующая итерации 2. Заметим, что те из выделенных ранее векторов условий, которые находятся в базисе последней основной таблицы, автоматически остаются выделенными. Дальнейшее течение процесса решения идет так же, как и в первой итерации. В качестве исходной во второй итерации принимается последняя основная таблица итерации 1. На протяжении этой итерации заполняется часть табл. II, соответствующая выделенным векторам условий (для определенно-

сти будем считать такими векторами A_1, A_j, A_n). Итерации проводятся до тех пор, пока одна из них не завершится либо альтернативой I, либо случаем 2⁰ альтернативы II. В первом случае процесс заканчивается получением искомого решения, во втором — установлением неразрешимости исследуемой задачи.

Для того чтобы иметь возможность применить метод уточнения оценок, необходимо определить предварительно некоторый опорный план сопряженной задачи. Что касается метода сокращения невязок, то его, как мы видели, можно применять, отправляясь от произвольного плана двойственной задачи. Это обстоятельство является достоинством рассматриваемого метода, поскольку во многих случаях выбор неопорного плана сопряженной задачи производится без всяких вычислений. Если же определение исходного плана связано с затруднениями, следует обратиться к методу, изложенному в п. 2.5 настоящей главы. Задача V в этом случае дополняется соотношением (4.68)

$$\sum_{i=1}^n x_i + x_0 = M,$$

где $x_0 \geq 0$; M — достаточно большое число.

В результате образуется новая задача линейного программирования V' . Как было отмечено, в качестве плана задачи VI' [(4.69), (4.70)], сопряженной с V' , можно принять вектор $Y' = (y_0', 0, 0, \dots, 0)$, где

$$y_0' = \max_{1 \leq j \leq n} \{c_j, 0\}.$$

Решим теперь задачу V' методом сокращения невязок, отправляясь от плана Y' (в процессе решения M предполагается достаточно большим). Пусть $\bar{X}^* = (\bar{x}_0^*, \bar{x}_1^*, \bar{x}_2^*, \dots, \bar{x}_n^*)$ — оптимальный план задачи V' . Возможны два случая:

1) в последней строке табл. II под вектором A_0 стоит нуль,

2) в этом месте таблицы расположена положительная величина.

В первом случае искомым оптимальным планом задачи V является вектор $X^* = (x_1^*, x_2^*, \dots, x_n^*)$. Во

втором — задача V неразрешима. Справедливость приведенного правила была доказана в п. 2.5 (в соответствии с алгоритмом рассматриваемого метода в указанной позиции табл. II расположена компонента y_0^* оптимального плана задачи VI').

3.4. Пример. Решим методом сокращения невязок следующую задачу.

Максимизировать

$$L(x) = x_1 + x_2 + x_3 + 3x_4 + x_5 + 3x_6 + x_7$$

при условиях:

$$\begin{aligned} x_1 + 2x_2 + 2x_3 + 3x_4 + 2x_5 + 4x_6 + 2x_7 &= 5, \\ 2x_1 + x_2 + 4x_3 - x_4 + x_5 + 2x_6 - 3x_7 &= 7, \\ x_1 + 3x_2 + x_3 + 4x_4 - x_5 + 3x_6 + 2x_7 &= 5, \\ -x_1 + 2x_2 + x_3 + x_4 + 2x_5 - x_6 + x_7 &= 2, \\ x_j &\geq 0, \quad j = 1, 2, 3, 4, 5, 6, 7. \end{aligned}$$

Все коэффициенты первого ограничения положительны. Следовательно, исходный план сопряженной задачи определяется очень просто

$$y_1 = \max \left\{ \frac{1}{1}, \frac{1}{2}, \frac{1}{2}, \frac{3}{3}, \frac{1}{2}, \frac{3}{4}, \frac{1}{2} \right\} = 1;$$

$$y_2 = y_3 = y_4 = 0.$$

Составляем табл. II. В строку (δ) вписываются величины

$$\delta_j = \sum_{i=1}^4 a_{ij} y_i - c_j, \quad j = 1, 2, 3, 4, 5, 6, 7.$$

Нулевые значения δ_j (δ_1, δ_4) определяют векторы условий, которые участвуют в первой вспомогательной задаче A_1, A_4 . После этого заполняется первая основная таблица (кроме последнего столбца). Элементы последней строки табл. Ia выписываются в столбец μ табл. II. Затем определяются величины $\Delta_j, j = 1, 4$;

$$\Delta_1 = \sum_{i=1}^4 a_{i1} \mu_i = 1 + 2 + 1 - 1 = 3; \quad \Delta_4 = 7.$$

Поскольку среди элементов строки Δ имеются положительные, итерация 1 продолжается. $\max\{3, 7\} = 7$. Сле-

довательно, для ввода в базис B_2 выбирается вектор A_4 . Коэффициенты разложения A_4 по базису B_1 , совпадающие с его компонентами, записываются в последний столбец табл. Ia.

Т а б л и ц а II

c	1	1	1	3	1	3	1							
A	A_1	A_2	A_3	A_4	A_5	A_6	A_7	μ	μ'	μ''	$\bar{\mu}$	$\bar{\bar{\mu}}$	$\bar{\bar{\bar{\mu}}}$	
1	1	2	2	3	2	4	2	1	1	1	-1	-4		
2	2	1	4	-1	1	2	-3	1	1	$\frac{4}{9}$	$\frac{2}{3}$	1		
3	1	3	1	4	-1	3	2	1	$-\frac{3}{4}$	$-\frac{8}{9}$	$\frac{2}{3}$	3		
4	-1	2	1	1	2	-1	1	1	1	1	1	1		
δ	$\times$	1	1	$\times$	1	1	1	I						
Δ	3			7										
Δ'	$\frac{5}{4}$													
Δ''		$\frac{16}{9}$	$\frac{35}{9}$		$\frac{16}{3}$	$\frac{11}{9}$	$-\frac{1}{9}$	$\theta_0 = \frac{3}{16}$	II					
$\bar{\delta}$	$\times$	$\frac{2}{3}$	$\frac{13}{48}$	$\times$	$\times$	$\frac{37}{48}$	$\frac{49}{48}$							
$\bar{\Delta}$		$\frac{8}{3}$	$\frac{7}{3}$			$-\frac{5}{3}$	$-\frac{5}{3}$	$\bar{\theta}_0 = \frac{13}{102}$						
$\bar{\bar{\delta}}$	$\times$	$\frac{5}{14}$	$\times$	$\times$	$\times$	$\frac{9}{4}$	$\frac{17}{6}$	III						
$\bar{\bar{\Delta}}$		4			-8	-6	-4							$\bar{\bar{\theta}}_0 = \frac{5}{56}$
$\bar{\bar{\bar{\delta}}}$	$\times$	$\times$	$\times$	$\times$	$\frac{5}{7}$	$\frac{3}{2}$	$\frac{11}{7}$							
δ^*	0	0	0	0	$\frac{5}{7}$	$\frac{3}{2}$	$\frac{11}{7}$	IV						

Таблица 1а

c	e B_1	e_0	e_1	e_2	e_3	e_4	A_1
1	e_1	5	1				3
1	e_2	7		1			-1
← 1	e_3	5			1		4
1	e_4	2				1	1
μ		19	1	1	1	1	7

В правый нижний угол табл. 1а помещается $\Delta_4 = 7$. Поскольку $\min\left\{\frac{5}{3}, \frac{5}{4}, \frac{2}{1}\right\} = \frac{5}{4}$ из базиса B_1 выводится вектор e_3 , что отмечается стрелкой, помещенной слева от соответствующей строки. Построение табл. 1а на этом заканчивается. Переходим к заполнению табл. 1б. 3-я строка табл. 1б образуется из 3-й строки табл. 1а (эта строка выделена) путем деления всех ее элементов на последний (4). Для получения i -й строки табл. 1б ($i \neq 3$) необходимо из соответствующей строки табл. 1а вычесть 3-ю строку табл. 1б, умноженную на элемент последнего столбца табл. 1а, стоящий на i -м месте.

При этом последний столбец табл. 1б остается пока незаполненным. Далее, возвращаясь к табл. II, вычисляем элементы строки Δ' : $\Delta'_1 = \frac{5}{4}$; $\Delta'_4 = 0$. Поскольку положительным оказался лишь Δ'_1 в новый базис B_3 следует ввести вектор A_1 . Коэффициенты разложения A_1 по базису B_2 находятся по формулам

$$z_{i1} = \sum_{t=1}^4 e'_{it} a_{t1}, \quad i = 1, 2, 3, 4.$$

Другими словами, для получения z_{i1} необходимо элементы i -й строки табл. 1б умножить на соответствующие координаты вектора A_1 и результаты сложить. В правый нижний угол табл. 1б помещается $\Delta'_1 = 5/4$.

Таблица 16

c	e B_2	e_0	e_1	e_2	e_3	e_4	A_1
1	e_1	$\frac{5}{4}$	1		$-\frac{3}{4}$		$\frac{1}{4}$
← 1	e_2	$\frac{33}{4}$		1	$\frac{1}{4}$		$\frac{9}{4}$
→	A_4	$\frac{5}{4}$			$\frac{1}{4}$		$\frac{1}{4}$
1	e_4	$\frac{3}{4}$			$-\frac{1}{4}$	1	$-\frac{5}{4}$
μ'		$\frac{41}{4}$	1	1	$-\frac{3}{4}$	1	$\frac{5}{4}$

Сравнивая между собой величины $\frac{e_{i0}}{z_{i1}}$ при $z_{i1} > 0$, обнаруживаем, что удалению из базиса B_2 подлежит вектор e_2

$$\min \left\{ \frac{\frac{5}{4}}{\frac{1}{4}}, \frac{\frac{33}{4}}{\frac{9}{4}}, \frac{\frac{5}{4}}{\frac{1}{4}} \right\} = \frac{33}{9}.$$

Таблица 1в

c	e B_3	e_0	e_1	e_2	e_3	e_4	A_5
← 1	e_1	$\frac{1}{3}$	1	$-\frac{1}{9}$	$-\frac{7}{9}$		$\frac{8}{3}$
→	A_1	$\frac{11}{3}$		$\frac{4}{9}$	$\frac{1}{9}$		$\frac{1}{3}$
	A_4	$\frac{1}{3}$		$-\frac{1}{9}$	$\frac{2}{9}$		$-\frac{1}{3}$
1	e_4	$\frac{16}{3}$		$\frac{5}{9}$	$-\frac{1}{9}$	1	$\frac{8}{3}$
μ''		$\frac{17}{3}$	1	$\frac{4}{9}$	$-\frac{8}{9}$	1	$\frac{16}{3}$

Теперь переходим от табл. Iб к табл. Iв, заполняя все ее столбцы, кроме последнего. Вычислив затем элементы строки Δ'' табл. II, убеждаемся, что первая вспомогательная задача решена: $\Delta_1'' = \Delta_4'' = 0$. Однако решение исследуемой задачи еще не получено, так как базис B_3 содержит единичные векторы e_1, e_4 . Следовательно, необходимо вычислить оставшиеся элементы строки Δ'' и с их помощью образовать строку $\bar{\delta}$, которая определит набор векторов условий новой вспомогательной задачи

$$\bar{\delta}_j = \delta_j - \theta_0 \Delta_j'',$$

где

$$\theta_0 = \min_{\Delta_j'' > 0} \left\{ \frac{\delta_j}{\Delta_j''} \right\}, \quad j = 1, 2, 3, 4, 5, 6, 7.$$

В результате выделяются векторы A_1, A_3, A_4 . Среди элементов строки Δ'' , соответствующих вновь выделенным векторам, имеется лишь один положительный Δ_5'' . Поэтому в базис B_4 следует включить вектор A_5 . Вторая итерация состоит всего из одного шага. Построив табл. Iг и заполнив строку $\bar{\Delta}$ табл. II, обнаруживаем, что вторая вспомогательная задача решена. Вместе с тем базис B_4 содержит вектор e_4 . Поэтому необходимо приступить к итерации 3. Эта итерация проводится также за один шаг. В результате получаем базис B_5 , содержащий векторы A_3, A_1, A_4, e_4 . Перед началом четвертой итерации выделенными оказываются векторы A_1, A_2, A_3, A_4 . Итерация 4 за один шаг приводит к выводу из базиса последнего из единичных векторов e_4 . Таким образом, табл. Iе определяет оптимальный план рассматриваемой задачи $X^* = (1, 1, 1, 0, 0, 0, 0)$. С помощью этой таблицы нетрудно также получить решение сопряженной задачи.

Т а б л и ц а Iг

с	e						
	B_4	e_0	e_1	e_2	e_3	e_4	A_3
→ ←	A_5	$\frac{1}{8}$	$\frac{3}{8}$	$-\frac{1}{24}$	$-\frac{7}{24}$		$\frac{7}{24}$
	A_1	$\frac{29}{8}$	$-\frac{1}{8}$	$\frac{11}{24}$	$\frac{5}{24}$		$\frac{43}{24}$

Продолжение табл. 1г

c	$\begin{matrix} e \\ B_4 \end{matrix}$	e_0	e_1	e_2	e_3	e_4	A_3
	A_4	$\frac{3}{8}$	$\frac{1}{8}$	$-\frac{1}{8}$	$\frac{1}{8}$		$\frac{1}{8}$
1	e_4	5	-1	$\frac{2}{3}$	$\frac{2}{3}$	1	$\frac{7}{3}$
$\overline{12}$		5	-1	$\frac{2}{3}$	$\frac{2}{3}$	1	$\frac{7}{3}$

Таблица 1д

c	$\begin{matrix} e \\ B_5 \end{matrix}$	e_0	e_1	e_2	e_3	e_4	A_2
$\rightarrow$	A_3	$\frac{3}{7}$	$\frac{9}{7}$	$-\frac{1}{7}$	-1		$-\frac{4}{7}$
	A_1	$\frac{20}{7}$	$-\frac{17}{7}$	$\frac{5}{7}$	2		$\frac{13}{7}$
	A_4	$\frac{3}{7}$	$\frac{2}{7}$	$-\frac{1}{7}$			$\frac{3}{7}$
$\leftarrow$	1	4	-4	1	3	1	4
$\overline{12}$		4	-4	1	3	1	4

Таблица 1е

c	$\begin{matrix} e \\ B_3 \end{matrix}$	e_0	e_1	e_2	e_3	e_4	
1	A_3	1	$\frac{5}{7}$		$-\frac{4}{7}$	$\frac{1}{7}$	
1	A_1	1	$-\frac{4}{7}$	$\frac{1}{4}$	$\frac{17}{28}$	$-\frac{13}{28}$	
3	A_4		$\frac{5}{7}$	$-\frac{1}{4}$	$-\frac{9}{28}$	$-\frac{3}{28}$	
$\rightarrow$ 1	A_2	1	-1	$\frac{1}{4}$	$\frac{3}{4}$	$\frac{1}{4}$	
$\overline{12}$							

Для этого слева от нее выписываются соответствующие коэффициенты линейной формы исходной задачи. После чего полученный столбец умножается на столбцы e_1, e_2, e_3, e_4 табл. 1е.

$$y_1^* = 1 \cdot \frac{5}{7} - 1 \cdot \frac{4}{7} + 3 \cdot \frac{5}{7} - 1 \cdot 1 = \frac{9}{7};$$

$$y_2^* = -\frac{1}{4}; \quad y_3^* = -\frac{5}{28}; \quad y_4^* = -\frac{11}{28}.$$

Итак,

$$Y^* = \left(\frac{9}{7}, -\frac{1}{4}, -\frac{5}{28}, -\frac{11}{28} \right).$$

Предоставляем читателю убедиться в том, что векторы X^* и Y^* действительно являются планами рассматриваемых двойственных задач, причем

$$\sum_{j=1}^7 c_j x_j^* = \sum_{i=1}^4 b_i y_i^*.$$

Это послужит проверкой правильности проведенных расчетов.

3.5. Об оценках быстроты сходимости метода сокращения невязок. При использовании метода сокращения невязок строится последовательность планов сопряженной задачи, монотонно сходящаяся к ее решению. Поэтому после проведения каждой из итераций метода можно легко получить ограничение сверху для оптимальной величины линейной формы исследуемой задачи. В некоторых случаях по решению вспомогательной задачи удастся образовать близкий к нему план исходной задачи. Это дает нижний предел для искомого оптимального плана.

Таким образом, создается возможность оценки результата каждой итерации метода сокращения невязок. Особенно просто двухсторонние оценки получаются в том случае, когда среди векторов условий задачи V имеется полный набор m -мерных единичных векторов. Другими словами, когда условия исследуемой задачи имеют вид

$$\sum_{j=1}^n a_{ij} x_j \leq b_i, \quad i = 1, 2, \dots, m,$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n,$$

причем $b_j \geq 0$ для всех j .

Нетрудно заметить, что при таких условиях каждый план любой из вспомогательных задач является планом исходной задачи. Поэтому решение подобных задач методом сокращения невязок приводит к переходу от одного опорного плана к другому. При этом с каждым планом связывается оценка его отклонения от оптимального (разность соответствующих значений линейных форм исходной и сопряженной задач).

Вообще говоря, план вспомогательной задачи не удовлетворяет всем условиям системы (4.28). Однако иногда его легко дополнить до некоторого плана задачи V. Подобная ситуация имеет место, например, для задачи транспортировки (§ 3 гл. 5). Следует отметить, что за счет небольшой модификации метода сокращения невязок можно и в общем случае двигаться по планам исследуемой задачи. Правда, для того чтобы начать такой процесс, необходимо иметь, кроме плана сопряженной задачи VI, опорный план исходной задачи V. Остановимся вкратце на общей схеме соответствующего алгоритма. Будем отправляться от опорного плана задачи V $X = (x_1, x_2, \dots, x_n)$ и плана сопряженной задачи $Y = (y_1, y_2, \dots, y_m)$. Под E_Y , как и прежде, понимается совокупность таких индексов j , что

$$\sum_{i=1}^m a_{ij} y_i = c_j.$$

Через E_X обозначим множество индексов тех векторов условий задачи V, которые входят в базис плана $X = (x_1, x_2, \dots, x_n)$. Пусть $E_{X,Y}$ — совокупность индексов, входящих в E_X , но не принадлежащих E_Y ; $\bar{E}_{X,Y}$ — множество индексов, содержащихся либо в E_X , либо в E_Y . С парой планов X, Y связывается вспомогательная задача, состоящая в минимизации линейной формы

$$\sum_{j \in E_{X,Y}} x_j \quad (4.103)$$

при условиях:

$$\sum_{j \in \bar{E}_{X,Y}} x_j A_j = B; \quad (4.104)$$

$$x_j \geq 0, \quad j \in \bar{E}_{X,Y}.$$

За исходный план задачи (4.103), (4.104) можно принять вектор X .

Если минимальное значение линейной формы (4.103) равно нулю, решение вспомогательной задачи совпадает с оптимальным планом задачи V [(4.27)—(4.29)]. Если это не так, то по решению вспомогательной задачи и вектору Y составляется новый план сопряженной задачи VI [(4.30), (4.31)], связанный с меньшим значением линейной формы (4.30). С помощью полученного плана задачи VI и оптимального плана предыдущей вспомогательной задачи формулируется новая вспомогательная задача (по уже изложенному правилу). В качестве ее исходного плана принимается решение предыдущей вспомогательной задачи, являющееся вместе с тем планом задачи V . Процесс продолжается до получения вспомогательной задачи с минимальным значением линейной формы, равным нулю. Решение этой задачи совпадает с оптимальным планом исследуемой задачи V .

Итак, каждая итерация намеченной здесь модификации метода сокращения невязок приводит к планам исходной и сопряженной задач V , VI , из которых второй естественно использовать для оценки отклонения первого от оптимума. Обоснование изложенного приема может быть проведено почти так же, как и для метода сокращения невязок (п. 3.2). Соответствующий алгоритм состоит в последовательном заполнении вспомогательной и основных таблиц по правилам, описанным в п. 3.3.

ГЛАВА 5

ТРАНСПОРТНАЯ ЗАДАЧА

Транспортная задача является одной из важных частных задач линейного программирования. Впервые, насколько нам известно, транспортная задача рассматривалась А. Н. Толстым [16]. В американской литературе первая работа, посвященная этой задаче [48], относится к 1941 г. и принадлежит Хичкоку, в честь которого задача эта иногда именуется в американской печати проблемой Хичкока.

Следует отметить, что в обеих упомянутых работах не было еще законченного метода решения рассматриваемой задачи. Первый общий метод решения транспортной проблемы, получивший в дальнейшем название «метода потенциалов», предложен Л. В. Канторовичем в 1949 г. [11]*). Метод Канторовича был разработан вне связи с общими методами линейного программирования. Несколько позже (1950 г.) тот же метод, исходя из общих соображений, был получен Данцигом [30].

Специфичность условий задачи транспортировки приводит к тому, что наряду с обычными свойствами задачи линейного программирования она обладает рядом интересных, свойственных лишь ей особенностей. В настоящей главе транспортная задача подробно изучается как в качественном, так и в количественном аспектах.

Особое внимание при этом уделяется методам решения этой задачи и связанным с ними алгоритмам. Для решения транспортной задачи применяются приемы,

*) Работа [11] была выполнена еще в 1940 г.

уже рассмотренные нами в связи с общей проблемой линейного программирования, — метод последовательного улучшения плана и метод последовательного сокращения невязок. Однако в применении к этой задаче они существенно упрощаются и приводят к компактным вычислительным схемам.

§ 1. ПОСТАНОВКА ВОПРОСА И ПРЕДВАРИТЕЛЬНЫЕ ЗАМЕЧАНИЯ

1.1. Постановка задачи. В этом параграфе дается постановка транспортной задачи и обсуждаются некоторые присущие ей особенности. В основном результаты параграфа носят качественный характер. Многие из них будут использованы в последующем при описании методов решения задачи транспортировки. Содержание транспортной задачи состоит в следующем. Пусть в пунктах $A_1, A_2, \dots, A_m$ производится некоторый однородный продукт, причем объем производства в пункте A_i составляет a_i единиц. Допустим также, что указанный продукт потребляется в пунктах $B_1, B_2, \dots, B_n$, причем объем потребления в пункте B_j составляет b_j единиц продукта. Пункты $A_1, \dots, A_m$ будем называть пунктами производства, пункты $B_1, B_2, \dots, B_n$ — пунктами потребления. В задаче предполагается, что суммарный объем производства $(\sum_{i=1}^m a_i)$ совпадает с суммарным объемом потребления $(\sum_{j=1}^n b_j)$. Положим, что из каждого пункта производства возможна транспортировка продукта в любой из пунктов потребления, причем транспортные издержки, приходящиеся на единицу продукта при перевозке его из i -го пункта производства (A_i) в j -й пункт потребления (B_j), заданы и составляют c_{ij} денежных единиц. Задача состоит в составлении такого плана перевозок, при котором запросы всех пунктов потребления были бы удовлетворены за счет производства в пунктах $A_1, A_2, \dots, A_m$ и при этом суммарные транспортные издержки были бы минимальными. Все условия транспортной задачи удобно записать в виде наглядной табл. 5.1.

Т а б л и ц а 5.1

		Пункты потребления				
		B_1	$B_2 \dots B_{n-1}$	B_n		
Пункты производства	A_1	c_{11}	$c_{12} \dots c_{1, n-1}$	c_{1n}	a_1	Объем производства
	A_2	c_{21}	$c_{22} \dots c_{2, n-1}$	c_{2n}	a_2	
	.	.	.	.	.	
	.	.	.	.	.	
	A_m	c_{m1}	$c_{m2} \dots c_{m, n-1}$	c_{mn}	a_m	
		b_1	$b_2 \dots b_{n-1}$	b_n	Объем потребления	

Табл. 5.1 заполнена величинами транспортных издержек. В нижней строке помещены объемы потребления пунктов $B_1, B_2, \dots, B_n$, в последнем столбце — объемы производства в пунктах $A_1, A_2, \dots, A_m$.

Приведем теперь строгую постановку задачи транспортировки. Пусть x_{ij} обозначает количество продукта, перевозимое из A_i в B_j (величину перевозки из A_i в B_j).

Требуется определить совокупность величин x_{ij} , $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$ (план перевозок), удовлетворяющих условиям:

$$\left. \begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, \quad i = 1, 2, \dots, m; \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = 1, 2, \dots, n; \end{aligned} \right\} \tag{5.1}$$

$$x_{ij} \geq 0; \tag{5.2}$$

и такую, что линейная форма

$$L(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \tag{5.3}$$

(суммарные транспортные издержки) достигает на ней своего минимума, т. е. для любой другой совокупности $X' = \{x'_{ij}\}$, удовлетворяющей условиям (5.1), (5.2), имеет место соотношение

$$L(X) \leq L(X').$$

Очевидно, что если величины x_{ij} удовлетворяют условиям (5.1),

$$\sum_{i=1}^m a_i = \sum_{i=1}^m \sum_{j=1}^n x_{ij} = \sum_{j=1}^n \sum_{i=1}^m x_{ij} = \sum_{j=1}^n b_j,$$

т. е. равенство суммарного объема производства суммарному объему потребления

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j \quad (5.4)$$

необходимо для разрешимости задачи (5.1), (5.2), (5.3). Ниже будет показано, что (5.4) является вместе с тем и достаточным условием для разрешимости рассматриваемой задачи. Заметим также, что из условий (5.2) вытекает неотрицательность величин a_i , b_j . С другой стороны, если $a_i = 0$ ($b_j = 0$), то пункт A_i (B_j) можно исключить из рассмотрения. Поэтому естественно считать

$$a_i > 0, \quad i = 1, \dots, m; \quad b_j > 0, \quad j = 1, \dots, n. \quad (5.5)$$

Сформулированную задачу (5.1), (5.2), (5.3), величины a_i и b_j которой удовлетворяют естественным условиям (5.4), (5.5), будем кратко обозначать через T . Переменные x_{ij} задачи T связаны посредством $n + m$ линейных уравнений. Легко заметить, что любое из этих уравнений является следствием остальных и поэтому может быть отброшено. Проверим высказанное утверждение, например, для уравнения

$$\sum_{j=1}^n x_{1j} = a_1. \quad (5.6)$$

Пусть совокупность величин x_{ij} удовлетворяет оставшимся $n + m - 1$ уравнениям системы (5.1) [исключая (5.6)]. Тогда совокупность $\{x_{ij}\}$ удовлетворяет также и уравнению (5.6). Действительно,

$$\sum_{j=1}^n x_{1j} = \sum_{i=1}^m \sum_{j=1}^n x_{ij} - \sum_{i=2}^m \sum_{j=1}^n x_{ij} = \sum_{j=1}^n \sum_{i=1}^m x_{ij} - \sum_{i=2}^m \sum_{j=1}^n x_{ij}.$$

Но в соответствии с уравнениями системы (5.1), которым по условию удовлетворяют величины x_{ij} ,

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, \dots, n; \quad \sum_{j=1}^n x_{ij} = a_i, \quad i = 2, \dots, m.$$

Поэтому

$$\sum_{j=1}^n x_{ij} = \sum_{j=1}^n b_j - \sum_{i=2}^m a_i.$$

Учитывая далее равенство (5.4), окончательно получаем

$$\sum_{j=1}^n x_{1j} = a_1.$$

Таким образом, система (5.1) содержит не более чем $n + m - 1$ независимых уравнений. В дальнейшем мы увидим, что любые $n + m - 1$ уравнений системы (5.1) независимы (ранг системы равен $n + m - 1$). Несмотря на зависимость любого из уравнений (5.1) от остальных, при формулировке транспортной задачи система (5.1) из соображений симметрии выписывается полностью. Итак, транспортная проблема является задачей линейного программирования с nm переменными и $n + m - 1$ условиями-равенствами. Задача T записана в канонической форме. Иногда бывает удобно записывать условия (5.1) в векторной форме. Обозначим через P_{ij} $(n + m)$ -мерный вектор, i -я и $(m + j)$ -я компоненты которого равны 1, а остальные — нули:

$$P_{ij} = \begin{pmatrix} 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \\ 1 \\ 0 \\ \vdots \\ 0 \end{pmatrix} \begin{matrix} \downarrow \\ i - 1 \\ \uparrow \\ \downarrow \\ m + j - i \\ \uparrow \\ \downarrow \\ n - j + 1 \\ \uparrow \end{matrix}.$$

Каждый такой вектор соответствует коммуникации, связывающей некоторый пункт производства с некоторым пунктом потребления. В частности, вектор P_{ij} соответствует коммуникации между A_i и B_j . В связи с этим векторы P_{ij} естественно называть векторами коммуникаций. Введем также вектор

$$P = \begin{pmatrix} a_1 \\ a_2 \\ \vdots \\ a_m \\ b_1 \\ b_2 \\ \vdots \\ b_n \end{pmatrix},$$

который в дальнейшем будем называть вектором производства — потребления. Теперь условия (5.1) могут быть переписаны в компактной форме

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} P_{ij} = P. \quad (5.1')$$

Если пользоваться терминологией общей задачи линейного программирования, то векторы P_{ij} следует называть векторами условий, P — вектором ограничений. Поскольку векторы условий транспортной задачи P_{ij} имеют чрезвычайно простую структуру, любой из методов линейного программирования применительно к этой задаче существенно упрощается. В этом мы убедимся в последующих параграфах настоящей главы.

Введем несколько важных понятий, связанных с транспортной задачей.

Совокупность величин c_{ij} , записанную в виде прямоугольной матрицы (таблицы)

$$C = \begin{vmatrix} c_{11} & c_{12} \dots c_{1n} \\ c_{21} & c_{22} \dots c_{2n} \\ \cdot & \cdot \cdot \cdot \cdot \\ \cdot & \cdot \cdot \cdot \cdot \\ c_{m1} & c_{m2} \dots c_{mn} \end{vmatrix} = \|c_{ij}\|_{m,n},$$

будем называть матрицей транспортных издержек.

Любую совокупность величин x_{ij} , удовлетворяющую условиям (5.1), (5.2), назовем планом перевозок задачи T , или коротко — планом. План задачи T удобно записывать в виде матрицы

$$X = \begin{vmatrix} x_{11} & x_{12} \dots x_{1n} \\ \cdot & \cdot \cdot \cdot \cdot \\ \cdot & \cdot \cdot \cdot \cdot \\ x_{m1} & x_{m2} \dots x_{mn} \end{vmatrix} = \|x_{ij}\|_{n,m},$$

состоящей из m строк и n столбцов.

План перевозок, обращающий в минимум суммарные транспортные издержки, называется оптимальным планом или решением задачи транспортировки (коротко — решением).

Будем говорить, что план X опорный, если система векторов P_{ij} , соответствующая всем тем коммуникациям,

по которым согласно X намечены положительные перевозки, линейно независима *). Систему векторов опорного плана, соответствующих положительным перевозкам, назовем базисом этого плана. Заметим, что последние четыре определения формулировались в гл. I применительно к общей задаче линейного программирования.

Понятие опорного плана в случае транспортной задачи имеет довольно наглядный смысл, вытекающий из следующего утверждения.

Теорема 1. Для опорности плана перевозок необходимо и достаточно, чтобы из коммуникации с положительными перевозками невозможно было бы составить замкнутый маршрут.

Доказательство этой теоремы приводится в § 3 гл. VI.

Поясним утверждение теоремы I примером. Пример, в частности, поясняет смысл понятия замкнутого маршрута. Допустим, что имеются три пункта производства с объемами 2, 1, 3 и три пункта потребления с объемами 4, 1, 1. На рис. 5.1 и 5.2 намечены два плана перевозок.

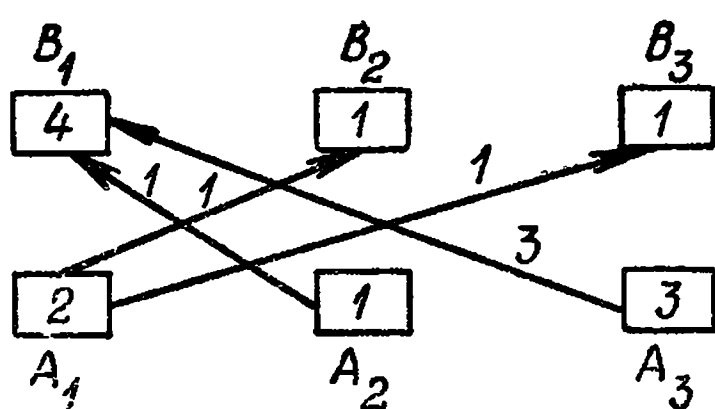


Рис. 5.1. План I.

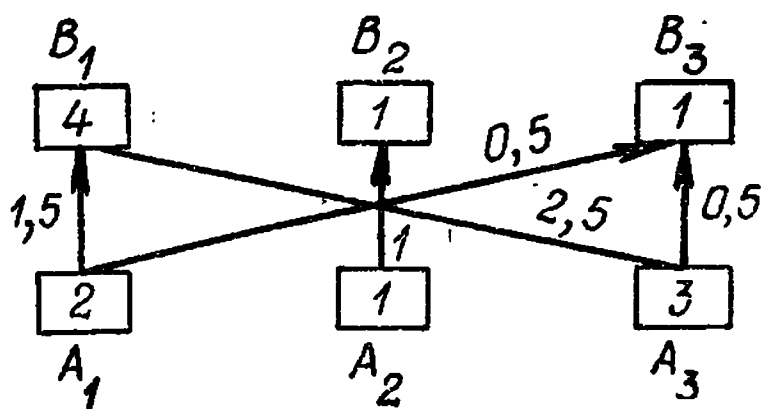


Рис. 5.2. План II.

Согласно первому плану из A_1 перевозится 1 единица в B_2 и 1 единица в B_3 ; из A_2, A_3 перевозится 4 имеющиеся там единицы в B_1 . Очевидно, что из намеченных стрелками коммуникаций невозможно составить ни одного замкнутого маршрута. Поэтому план I опорный. Обратимся теперь ко второму плану.

Двигаясь по отмеченным стрелками коммуникациям от A_1 к B_1 , от B_1 к A_3 , от A_3 к B_3 и, наконец, от B_3 к A_1 , получаем замкнутый маршрут. Следовательно, план II не является опорным. Это можно проверить и непосредственно. Действительно,

*) Определение линейной независимости векторов см. в п. 1.1 гл. 6.

**) Это определение относится к невырожденной задаче.

$$\begin{aligned}
 & P_{11} - P_{31} + P_{33} - P_{13} = \\
 & = \begin{pmatrix} 1 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \end{pmatrix} - \begin{pmatrix} 0 \\ 0 \\ 1 \\ 1 \\ 0 \\ 0 \end{pmatrix} + \begin{pmatrix} 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 1 \end{pmatrix} - \begin{pmatrix} 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 1 \end{pmatrix} = \begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{pmatrix} = 0,
 \end{aligned}$$

т. е. система векторов коммуникаций, соответствующих положительным перевозкам плана II, линейно зависима.

1.2. О разрешимости транспортной проблемы. Как уже отмечалось в предыдущих главах, встречаются случаи, когда задача линейного программирования оказывается неразрешимой. Поэтому, прежде чем формулировать какую-либо задачу линейного программирования, необходимо убедиться в том, что ее постановка имеет смысл. Ниже приводится условие, гарантирующее разрешимость транспортной проблемы.

Теорема 2. *Транспортная задача (5.1), (5.2), (5.3), в которой суммарный объем производства совпадает с суммарным объемом потребления*

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j, \quad (5.4)$$

имеет решение.

Доказательство. В силу теоремы 1 § 8 гл. 3, устанавливающей условия разрешимости общей задачи линейного программирования, для доказательства сформулированного утверждения достаточно показать, что:

а) существует по крайней мере один план задачи (5.1), (5.2), (5.3);

б) линейная форма (5.3) ограничена на множестве планов данной задачи.

а) Положим $p = m + n$, где $m(n)$ — количество пунктов производства (потребления) рассматриваемой задачи.

Воспользуемся методом индукции.

Если $p = 2$, и условие (5.4) выполнено, утверждение а) очевидно. Предположим, что а) справедливо для произвольной транспортной задачи, удовлетворяющей условию (5.4), с общим числом пунктов производства и потребления $p \leq k - 1$. Пусть теперь $n + m = k$. По-

ложим $x_{11} = \min(a_1, b_1)$. Рассмотрим три возможных случая.

1. $x_{11} = a_1 < b_1$.

Положим $x_{1j} = 0, j = 2, 3, \dots, n$, и для определения остальных перевозок плана получим следующую систему условий:

$$\sum_{j=1}^n x_{ij} = a_i = a'_{i-1}, \quad i = 2, 3, \dots, m;$$

$$\sum_{i=2}^m x_{ij} = b'_j = \begin{cases} b_1 - a_1, & j = 1 \\ b_j, & j = 2, 3, \dots, n. \end{cases}$$

Очевидно, что

$$\sum_{i=1}^{m-1} a'_i = \sum_{j=1}^n b'_j. \quad (5.7)$$

Поэтому построение плана задачи транспортировки с числом пунктов, равным $k = n + m$, свелось к определению плана для задачи с числом $p = k - 1 = n + m - 1$, существование которого гарантируется предположением индукции и условием (5.7).

2. $x_{11} = b_1 < a_1$.

Полагая $x_{i1} = 0, i = 2, 3, \dots, m$, мы аналогично предыдущему случаю уменьшаем число p на единицу, сохраняя условие (5.4).

3. $x_{11} = b_1 = a_1$.

Положим

$$x_{i1} = x_{j1} = 0, \quad i = 2, 3, \dots, m; \quad j = 2, 3, \dots, n.$$

Тогда, как нетрудно заметить, остальные перевозки составляют план транспортной задачи, удовлетворяющей условию баланса (5.4), с числом $p = (m - 1) + (n - 1) = k - 2$. Итак, справедливость а) установлена для любой транспортной задачи. Что касается условия б), то проверка его проводится без труда.

Действительно, в силу (5.1), (5.2) $0 \leq x_{ij} \leq a_i$. Поэтому

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \geq \sum_{i=1}^m \sum_{j=1}^n \min(0, c_{ij}) a_i.$$

Теорема полностью доказана.

Таким образом, условие баланса $\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$

является необходимым и достаточным для разрешимости транспортной задачи.

Заметим, что метод доказательства пункта а) теоремы 2 носит, по существу, конструктивный характер и без труда может быть преобразован в алгоритм построения плана задачи транспортировки.

1.3. Алгоритм построения плана. План транспортной задачи, определяемый условиями (5.1), (5.2), удобно записывать в виде таблицы:

Ниже указывается способ последовательного ее заполнения. Алгоритм построения плана состоит из нескольких шагов. На каждом шаге заполняется либо строка таблицы (случай 1), либо столбец (случай 2), либо и то и другое (случай 3). Опишем отдельный шаг алгоритма.

$x_{11} x_{12} \dots x_{1n}$	a_1
$x_{21} x_{22} \dots x_{2n}$	a_2
$\vdots \vdots \vdots \vdots \vdots$	$\vdots$
$x_{m1} x_{m2} \dots x_{mn}$	a_m
$b_1 \ b_2 \dots b_n$	

Выбираем левый верхний элемент таблицы из числа еще не определенных, скажем $x_{i_\lambda j_\mu}$.

Полагаем

$$x_{i_\lambda j_\mu} = \min(\bar{a}_{i_\lambda}, \bar{b}_{j_\mu}),$$

где

$$\bar{a}_{i_\lambda} = a_{i_\lambda} - \sum_{j=1}^{j_\mu-1} x_{i_\lambda j}; \quad \bar{b}_{j_\mu} = b_{j_\mu} - \sum_{i=1}^{i_\lambda-1} x_{ij_\mu}.$$

Если $\bar{a}_{i_\lambda} < \bar{b}_{j_\mu}$, заполняем i_λ -ю строку таблицы, начиная с $(j_\mu + 1)$ -го элемента, нулями. Если $\bar{a}_{i_\lambda} > \bar{b}_{j_\mu}$, заполняем нулями j_μ -й столбец, начиная с $(i_\lambda + 1)$ -го элемента.

Если же $\bar{a}_{i_\lambda} = \bar{b}_{j_\mu}$, полагаем равными нулю еще неопределенные элементы i_λ -й строки и j_μ -го столбца таблицы. Затем переходим к следующему шагу.

Характерным для рассмотренного алгоритма является особый способ заполнения таблицы, составляющей искомый план задачи транспортировки: каждый шаг начинается с левого верхнего (северо-западного) элемента незаполненной части таблицы. Поэтому описанный

алгоритм принято называть методом северо-западного угла.

Пример 1.

Таблица 5.2'

1	0	0	0	1	0	0	0	0	0
2	0	0	0	2	2	0	0	0	0
2	1	0	0	3	3	3	1	0	0
0	0	2	2	4	4	4	4	4	0
5	1	2	2						
4	1	2	2						
2	1	2	2						
0	1	2	2						
0	0	2	2						
0	0	0	0						

Для наглядности после каждого шага полезно справа от таблицы выписывать неудовлетворенные потребности, снизу — количества продукта, невывезенные из пунктов производства. Построение плана заканчивается после того, как справа и снизу от таблицы остаются одни нули.

Шаг 1. Построение плана начинается с элемента $x_{11} = \min(a_1, b_1)$. Поскольку $a_1 = 1 < b_1 = 5$, имеет место случай 1; первая строка таблицы, начиная со второго элемента, заполняется нулями. В результате введения перевозки $x_{11} = 1$ продукт из пункта производства A_1 оказывается полностью вывезенным; потребности пункта B_1 уменьшаются до четырех единиц. Остальные пункты задачи эта перевозка не затрагивает. Таким образом, количества невывезенного продукта из пунктов производства A_1, A_2, A_3, A_4 измеряются теперь числами 0, 2, 3, 4 (они выписываются справа от таблицы в шестом столбце); потребности пунктов B_1, B_2, B_3, B_4 равны соответственно 4, 1, 2, 2 (они выписываются в шестой строке таблицы).

Шаг 2. Левым верхним элементом незаполненной части таблицы оказывается x_{21} . В соответствии с алгоритмом $x_{21} = \min(4, 2) = 2$. Поскольку снова имеет место случай 1 ($2 < 4$), строка 2 дополняется нулями. Количество оставшегося продукта в пунктах производства и неудовлетворенные потребности пунктов потреб-

ления выписываются в столбец 7 и строку 7 табл. 5.2 соответственно. Описанный процесс продолжается до получения нулевой строки и нулевого столбца (весь продукт вывезен, все пункты потребления удовлетворены). Последовательность действий, связанная с построением плана, полностью отражена в табл. 5.2.

Метод северо-западного угла обладает одним полезным свойством: план, вычисленный с его помощью, всегда оказывается опорным. Поскольку это обстоятельство используется в следующем параграфе, приведем соответствующее доказательство. Воспользуемся методом индукции по числу $p = m + n$.

При $p = 2$ утверждение очевидно. Пусть оно верно для всех $p \leq k - 1$. Покажем, что отсюда вытекает его справедливость для $p = k$. Итак, положим $p = m + n = k$. В соответствии с алгоритмом в систему векторов P_{ij} , для которых $x_{ij} > 0$, обязательно входит P_{11} . Обозначим через R совокупность всех остальных векторов этой системы (исключая P_{11}).

Пусть на первом шаге имел место случай 1 (рассмотрение остальных случаев проводится идентично).

Тогда согласно алгоритму у любого вектора $P_{i_\lambda j_\mu} \in R$ первый индекс $i_\lambda \neq 1$. Допустим теперь, что при некоторых $c_0, c_1, \dots, c_s$ справедливо равенство

$$c_0 P_{11} + \sum_{i=1}^s c_i P_i = 0, \quad (5.8)$$

где $P_1, P_2, \dots, P_s$ составляют систему R (для упрощения записи двойная индексация в обозначениях этих векторов опущена). Покажем, что в таком случае

$$c_0 = c_1 = \dots = c_s = 0.$$

В самом деле, первая координата любого из векторов R , как мы видели, нуль, первая же координата P_{11} равна единице. Поэтому $c_0 = 0$ и соотношение (5.8) эквивалентно следующему равенству:

$$\sum_{i=1}^s c_i P_i = 0. \quad (5.9)$$

Но по предположению индукции векторы $P_1, P_2, \dots, P_s$ линейно независимы (система R' является базисом плана транспортной задачи с числом $p = k - 1$). Следовательно,

$$c_1 = c_2 = \dots = c_s = 0.$$

Итак, система векторов P_{ij} , соответствующих положительным перевозкам рассматриваемого плана, линейно независима. Тем самым опорность плана, построение которого осуществляется методом северо-западного угла, установлена для произвольного числа $p = m + n$.

1.4. Транспортная задача с нарушенным балансом производства и потребления. До сих пор проблема транспортировки изучалась в предположении баланса производства и потребления, который, как мы видели, необходим и достаточен для разрешимости задачи (5.1), (5.2), (5.3). В практических задачах условие (5.4) выполняется не всегда

Допустим, например, что необходимо оптимальным образом распределить уголь, добываемый в ряде районов, между некоторыми районами-потребителями. Очевидно, что суммарный объем добычи угля обычно не совпадает с суммарными потребностями. Если объем добычи меньше объема потребления, полное удовлетворение всех районов-потребителей невозможно. В том же случае, когда объем добычи превышает суммарные потребности, возникает задача об оптимальном снабжении районов-потребителей, включающая в себя, в частности, рациональный выбор количества угля, вывозимого из каждого района добычи.

Дадим точную постановку задачи транспортировки с нарушенным балансом между производством и потреблением. Имеются m пунктов производства однородного продукта с объемами $a_1, a_2, \dots, a_m$ и n пунктов потребления этого продукта с объемами $b_1, b_2, \dots, b_n$. Предполагается, что

$$\sum_{i=1}^m a_i > \sum_{j=1}^n b_j.$$

Стоимость перевозки единицы продукта из i -го пункта производства в j -й пункт потребления составляет c_{ij} . Требуется найти такую совокупность величин x_{ij} , $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$ (план перевозок), что

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad (i = 1, 2, \dots, m; \text{вывозится не более того, что производится}),$$

$\sum_{i=1}^m x_{ij} = b_j,$ (потребности каждого пункта удовлетворяются полностью);
 $j = 1, 2, \dots, n;$
 $x_{ij} \geq 0$ (перевозки неотрицательны),
 и при этом линейная форма

$$L(X) = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

(суммарные транспортные издержки) достигает своего минимального значения. Сформулированную задачу обозначим кратко буквой $\tilde{T}$.

Наряду с задачей $\tilde{T}$ рассмотрим обычную транспортную задачу T , определяемую матрицей транспортных издержек $\|\bar{c}_{ij}\|_{m, n+1}$, с объемами производства и потребления, равными соответственно $\bar{a}_1, \bar{a}_2, \dots, \bar{a}_m; \bar{b}_1, \bar{b}_2, \dots, \bar{b}_{n+1}$, где

$$\bar{c}_{ij} = \begin{cases} c_{ij}, & j \neq n+1 \\ 0, & j = n+1 \end{cases}, \quad \bar{a}_i = a_i;$$

$$\bar{b}_j = \begin{cases} b_j, & j \neq n+1 \\ \sum_{i=1}^m a_i - \sum_{j=1}^n b_j, & j = n+1. \end{cases}$$

Очевидно,

$$\sum_{j=1}^{n+1} \bar{b}_j = \sum_{i=1}^m \bar{a}_i,$$

вследствие чего введенная в рассмотрение транспортная проблема действительно является задачей типа T . Пусть $\bar{X} = \|\bar{x}_{ij}\|_{m, n+1}$ — оптимальное решение этой задачи. Тогда $X = \|x_{ij}\|_{m, n}$, где $x_{ij} = \bar{x}_{ij}$, $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$, является оптимальным планом задачи $\tilde{T}$. В самом деле,

$$\sum_{j=1}^n x_{ij} = \sum_{j=1}^{n+1} \bar{x}_{ij} - \bar{x}_{i, n+1} = a_i - \bar{x}_{i, n+1} \leq a_i,$$

так как

$$\bar{x}_{i, n+1} \geq 0;$$

при $j \neq n + 1$

$$\sum_{i=1}^m x_{ij} = \sum_{i=1}^m \bar{x}_{ij} = \bar{b}_j = b_j.$$

Поэтому X — план задачи $\tilde{T}$. Заметим, что суммарные транспортные издержки планов $\bar{X}$ и X одинаковы. Далее, если бы существовал план задачи $\tilde{T}$, X' , связанный с меньшими транспортными издержками, то соответствующий ему план задачи T

$$\bar{X}' = \|\bar{x}'_{ij}\|_{m, n+1},$$

где

$$\bar{x}'_{ij} = \begin{cases} x'_{ij}, & j \neq n + 1, \\ a_i - \sum_{j=1}^n x'_{ij}, & j = n + 1, \end{cases}$$

приводил бы к меньшим транспортным издержкам по сравнению с оптимальным планом этой задачи.

Следовательно, план $X = \|x_{ij}\|_{m, n}$, соответствующий оптимальному плану задачи T , является решением задачи $\tilde{T}$. Итак, путем введения фиктивного пункта потребления с объемом

$$\sum_{i=1}^m a_i - \sum_{j=1}^n b_j$$

задача без труда сводится к обычной транспортной проблеме типа T .

Если суммарный объем производства $\sum_{i=1}^m a_i$ меньше суммарного объема потребления $\sum_{j=1}^n b_j$, то полное удовлетворение всех районов-потребителей невозможно. В этом случае естественно так организовать перевозки, чтобы наиболее важные районы удовлетворялись возможно полнее и вместе с тем общие транспортные издержки были бы минимальными. Подобная задача также может быть сведена к транспортной проблеме типа T .

1.5. Свойства решений транспортной задачи. Специфика условий (5.1) транспортной проблемы T приводит к интересным и важным качественным свойствам ее ре-

шений, описанием которых мы сейчас займемся. Матрица условий транспортной задачи A имеет вид

$$A = \| P_{11}, P_{12}, \dots, P_{1n}, P_{21}, P_{22}, \dots, P_{2n}, \dots, \\ P_{m1}, P_{m2}, \dots, P_{mn} \| = \\ = \left\| \begin{array}{cccccc} 1 & 1 \dots 1 & 0 & 0 \dots 0 \dots 0 & 0 \dots 0 & \\ 0 & 0 \dots 0 & 1 & 1 \dots 1 \dots 0 & 0 \dots 0 & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 \dots 0 & 0 & 0 \dots 0 \dots 1 & 1 \dots 1 & \\ 1 & 0 \dots 0 & 1 & 0 \dots 0 \dots 1 & 0 \dots 0 & \\ 0 & 1 \dots 0 & 0 & 1 \dots 0 \dots 0 & 1 \dots 0 & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ 0 & 0 \dots 1 & 0 & 0 \dots 1 \dots 0 & 0 \dots 1 & \end{array} \right\|.$$

В каждом ее столбце точно два элемента равны 1, а остальные — нули. Таким образом, структура матрицы чрезвычайно проста. Этим объясняется как существенное упрощение ряда методов линейного программирования при использовании их для решения транспортной задачи, так и качественные особенности задачи транспортировки. Сформулируем утверждение, устанавливающее важное свойство опорных планов задачи T .

Теорема 3. Пусть $X = \| x_{ij} \|_{m,n}$ — опорный план задачи T . В таком случае каждая его перевозка x_{ij} является линейной комбинацией величин $a_1, a_2, \dots, a_m; b_1, b_2, \dots, b_n$ (объемов производства и потребления) с коэффициентами 0, —1, 1.

Доказательство этой теоремы вынесено в § 3 гл. 6. В § 8 гл. 3 было показано, что любая разрешимая задача линейного программирования обладает опорным решением (теорема 2). Поэтому транспортная проблема T , разрешимость которой мы установили в предыдущем пункте, всегда имеет опорный оптимальный план, для которого, естественно, справедливо утверждение теоремы 3. Таким образом, у любой задачи T существует оптимальный план, все перевозки которого — линейные комбинации величин a_i, b_j с коэффициентами 0, —1, 1. Часто в транспортной задаче объемы производства и потребления во всех пунктах являются

целыми числами. К целочисленным условиям приводят и некоторые комбинаторные задачи, сводящиеся к транспортной (см. проблему выбора, изложенную в § 3 настоящей главы). Из только что сформулированного свойства решений транспортной проблемы с очевидностью вытекает важная для приложений теорема 4.

Теорема 4. *Если объемы производства и потребления задачи T — целые числа, то существует оптимальный план этой задачи, состоящий из целочисленных перевозок.* В качестве такого плана может быть принято любое опорное решение задачи.

Заметим, что в предположениях теоремы 4 можно гарантировать целочисленность не всех, а лишь некоторых решений транспортной задачи. Если задача транспортировки с целочисленными объемами производства и потребления имеет несколько решений, то среди них заведомо существуют нецелые. Действительно, пусть X_1, X_2 — оптимальные планы задачи T . Если среди перевозок X_1 или X_2 имеются нецелые, наше утверждение доказано.

Допустим, что $X_1 = \|x_{ij}^{(1)}\|$, $X_2 = \|x_{ij}^{(2)}\|$ — целочисленные планы. Поскольку планы X_1, X_2 не тождественны, найдутся такие индексы i, j , что $x_{ij}^{(1)} \neq x_{ij}^{(2)}$. Для определенности положим $x_{ij}^{(1)} > x_{ij}^{(2)}$. Введем в рассмотрение новый план задачи T

$$X_3 = \lambda X_1 + (1 - \lambda) X_2 = \|\lambda x_{ij}^{(1)} + (1 - \lambda) x_{ij}^{(2)}\|.$$

В соответствии со свойствами решений общей задачи линейного программирования (§ 5 гл. 1) X_3 — оптимальный план задачи T при любой величине $0 \leq \lambda \leq 1$. Рассмотрим перевозку, намеченную планом X_3 между A_i и B_j ,

$$x_{ij}^{(3)} = \lambda x_{ij}^{(1)} + (1 - \lambda) x_{ij}^{(2)}.$$

Если λ изменяется от 0 до 1, величина $x_{ij}^{(3)}$ заполняет отрезок между $x_{ij}^{(2)}$ и $x_{ij}^{(1)}$. В частности, при некотором λ_0

$$x_{ij}^{(3)}(\lambda_0) = \lambda_0 x_{ij}^{(1)} + (1 - \lambda_0) x_{ij}^{(2)} = x_{ij}^{(1)} - \frac{1}{2}.$$

Таким образом, при $\lambda = \lambda_0$ оптимальный план задачи T не является целочисленным.

1.6. Условия невырожденности транспортной задачи.

В начале этого параграфа было установлено, что среди уравнений системы (5.1) линейно независимых не более чем $n + m - 1$. На самом деле система (5.1) содержит ровно $n + m - 1$ независимых уравнений. Для того чтобы убедиться в этом, достаточно обнаружить квадратную подматрицу матрицы условий A порядка $n + m - 1$ с определителем, отличным от нуля. Как не трудно видеть, такой подматрицей является матрица, составленная из первых $n + m - 1$ компонент векторов $P_{1n}, P_{2n}, \dots, P_{mn}, P_{11}, P_{12}, \dots, P_{1n-1}$. Действительно, определитель этой подматрицы

$$\Delta = \begin{array}{c} \uparrow \\ m \\ \downarrow \\ \hline \uparrow \\ n-1 \\ \downarrow \end{array} \left| \begin{array}{ccc} 1 & 0 \dots 0 & 1 \dots 1 \\ 0 & 1 \dots 0 & 0 \dots 0 \\ 0 & 0 \dots 0 & 0 \dots 0 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0 & 0 \dots 1 & 0 \dots 0 \\ 0 & 0 \dots 0 & 1 \dots 0 \\ \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot \\ 0 & 0 \dots 0 & 0 \dots 1 \end{array} \right| = 1.$$

$\leftarrow m \rightarrow \leftarrow n-1 \rightarrow$

Поскольку любое из уравнений системы (5.1) является следствием остальных, а сама система содержит $n + m - 1$ независимых уравнений, произвольные $n + m - 1$ уравнений (5.1) линейно независимы. Таким образом, если отбросить одно из уравнений (5.1), транспортная задача может рассматриваться как частный случай общей задачи линейного программирования с nm переменными и $n + m - 1$ независимыми ограничениями типа равенства. В гл. 3 был выделен класс невырожденных задач линейного программирования. Как уже неоднократно отмечалось, необходимость такого выделения обуславливается тем, что методы линейного программирования в случае вырожденных задач требуют некоторой модификации. Подобное положение имеет место, например, для метода последовательного улучшения плана, подробно рассмотренного в гл. 3. Напомним определение невырожденности применительно к задаче транспортировки.

Транспортную задачу с m пунктами производства и n пунктами потребления назовем невырожденной, если базис любого ее опорного плана состоит из $n + m - 1$ векторов коммуникаций *). В случае транспортной задачи можно указать достаточно легко проверяемое условие ее невырожденности.

Для невырожденности задачи T необходимо и достаточно отсутствие такой неполной группы пунктов производства, суммарный объем производства которой точно совпадает с суммарными потребностями некоторой группы пунктов потребления. Другими словами, необходимое и достаточное условие невырожденности транспортной задачи состоит в том, что для любых двух систем индексов $i_1, i_2, \dots, i_t; j_1, j_2, \dots, j_s$, где $t + s < n + m$, имеет место неравенство

$$\sum_{k=1}^t a_{i_k} \neq \sum_{k=1}^s b_{j_k}. \quad (5.10)$$

Докажем это утверждение в ту и другую сторону.

Необходимость. Допустим, что задача T невырожденная, однако для некоторых систем индексов $i_1, i_2, \dots, i_t; j_1, j_2, \dots, j_s$ $t + s < n + m$ справедливо соотношение

$$\sum_{k=1}^t a_{i_k} = \sum_{k=1}^s b_{j_k}. \quad (5.11)$$

Разобьем пункты производства и потребления на две группы. К первой группе пунктов производства (группе производства) отнесем $A_{i_1}, A_{i_2}, \dots, A_{i_t}$. Ко второй — остальные пункты производства. Аналогично, в первую группу пунктов потребления (группу потребления) объединим $B_{j_1}, B_{j_2}, \dots, B_{j_s}$, во вторую — остальные пункты потребления. Равенство (5.11) означает, что объем производства первой группы производства совпадает с потребностями первой группы потребления. Поскольку

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j$$

аналогичное утверждение справедливо также относительно второй группы производства и второй груп-

*) Здесь под базисом понимается система векторов коммуникаций отвечающих положительным перевозкам данного опорного плана.

пы потребления. Пользуясь методом северо-западного угла, можно определить такие перевозки между пунктами первых групп в количестве не более чем $t+s-1$, что весь продукт, находящийся в пунктах производства 1-й группы, вывозится, причем запросы пунктов, входящих в первую группу потребления, удовлетворяются за счет этого продукта полностью. Точно также с помощью не более чем $(m-t) + (n-s) - 1$ перевозок можно вывезти весь продукт из пунктов 2-й группы производства, удовлетворив при этом все запросы пунктов 2-й группы потребления. Нетрудно видеть, что совокупность всех введенных перевозок образует план задачи T . Но поскольку общее число этих перевозок не превосходит $(t+s-1) + [(m-t) + (n-s) - 1] = n + m - 2$, получаем противоречие с предположением о невырожденности рассматриваемой задачи, которое и доказывает необходимость условия.

Достаточность. Пусть неравенство (5.10) имеет место для любых систем индексов $i_1, \dots, i_t; j_1, \dots, j_s$; $s+t < n+m$ и $X = \|x_{ij}\|_{n,m}$ — произвольный план исследуемой задачи T . Рассмотрим множество пунктов производства и потребления, которые с помощью коммуникаций с положительными (в соответствии с планом X) перевозками могут быть соединены с A_1 . Покажем, что это множество совпадает со всей совокупностью пунктов задачи T . Действительно, если бы оно состояло из $A_{i_1}, A_{i_2}, \dots, A_{i_t}; B_{j_1}, B_{j_2}, \dots, B_{j_s}$, где $t+s < n+m$, то согласно плану X каждый из пунктов $A_{i_k}, k=1, 2, \dots, t$ ($A_{i_1} = A_1$) вывозил бы свой продукт лишь в пункты $B_{j_1}, B_{j_2}, \dots, B_{j_s}$, причем за счет этих перевозок потребности последних были бы полностью удовлетворены. Таким образом, мы имели бы равенство

$$\sum_{k=1}^t a_{i_k} = \sum_{k=1}^s b_{j_k},$$

противоречащее условию (5.10).

Итак, пункт A_1 (а следовательно, и любой другой) можно соединить с помощью коммуникаций с положительными перевозками со всеми пунктами рассматриваемой транспортной задачи. Обозначим систему векторов коммуникаций, соответствующих ненулевым перевозкам

плана X , через G_X . Покажем, что любой вектор P_{ij} может быть представлен в виде линейной комбинации векторов G_X . Для этого соединим пункты A_i, B_j коммуникациями с ненулевыми перевозками:

$$\overline{A_i B_{j_1}}, \overline{A_{i_2} B_{j_1}}, \dots, \overline{A_{i_\lambda} B_{j_{\lambda-1}}}, \overline{A_{i_\lambda} B_j}.$$

По условию векторы $P_{i_k j_k}, P_{i_k j_{k-1}}, k=1, 2, \dots, \lambda (i_1=i, j_\lambda=j)$ принадлежат G_X . Тогда, как нетрудно усмотреть,

$$P_{ij} = P_{ij_1} - P_{i_2 j_1} + P_{i_2 j_2} - \dots - P_{i_\lambda j_{\lambda-1}} + P_{i_\lambda j}.$$

Таким образом, любой вектор P_{ij} представляется через векторы G_X . Но поскольку среди векторов P_{ij} имеются $n+m-1$ линейно независимых, система G_X также содержит $n+m-1$ линейно независимых векторов. Итак, система векторов, соответствующих положительным перевозкам произвольного плана X рассматриваемой задачи, всегда содержит $n+m-1$ линейно независимых векторов, что означает невырожденность данной транспортной проблемы. Достаточность условия доказана.

Будем называть план транспортной задачи X **нераспадающимся**, если любая пара пунктов этой задачи может быть соединена коммуникациями с положительными согласно X перевозками. В противном случае назовем X **распадающимся** планом. Из установленного критерия невырожденной транспортной проблемы, в частности, следует, что любой план невырожденной задачи T **нераспадающийся**. Действительно, из невырожденности T следует справедливость (5.10) (необходимость). Произвольный же план транспортной задачи, удовлетворяющей условию (5.10), как было показано при доказательстве достаточности критерия невырожденности, является **нераспадающимся**.

1.7. О задачах, сводящихся к транспортной. При исследовании транспортной задачи обычно оперируют такими терминами, как перевозка, коммуникация, транспортные издержки и т. п. Это, однако, вовсе не означает, что к задаче линейного программирования (5.1), (5.2), (5.3), которую мы называли транспортной, сводится лишь

проблема определения рациональных перевозок. В схему транспортной задачи укладывается целый ряд производственных задач.

Одной из таких задач является приводимая ниже проблема рационального использования станков.

Допустим, что имеется m различных станков $A_1, A_2, \dots, A_m$, каждый из которых может обрабатывать n различных деталей $B_1, B_2, \dots, B_n$. Известно, что станок A_i обрабатывает за единицу времени d_{ij} деталей типа B_j . Мы предположим, что производительность i -го станка при обработке j -й детали можно представить в виде $d_{ij} = \alpha_i \beta_j$, где величина α_i определяется станком, а β_j — видом обрабатываемой на нем детали. Пусть r_{ij} обозначает стоимость эксплуатации i -го станка по обработке деталей j -го типа в течение единицы времени. Задача заключается в таком рациональном распределении времени работы станков по обработке различных деталей, чтобы деталей типа B_j было обработано s_j штук, станок A_i работал не более h_i единиц времени и при этом стоимость обработки была бы минимальной. Математически эта задача формулируется так. Найти совокупность величин h_{ij} (времени работы i -го станка в j -м режиме), удовлетворяющих условиям:

$$\sum_{j=1}^n h_{ij} \leq h_i \quad (\text{ограничение работы } i\text{-го станка по времени});$$

$$i = 1, 2, \dots, m;$$

$$\sum_{i=1}^m \alpha_i \beta_j h_{ij} = s_j \quad (\text{деталей типа } B_j \text{ должно быть обработано } s_j \text{ штук});$$

$$j = 1, 2, \dots, n;$$

$$h_{ij} \geq 0 \quad (\text{время работы } i\text{-го станка в } j\text{-м режиме неотрицательно});$$

и такую, что суммарная стоимость эксплуатации станков $\sum_{i=1}^m \sum_{j=1}^n r_{ij} h_{ij}$ достигает на ней своего минимального значения.

Покажем, что поставленная задача сводится к задаче (5.1), (5.2), (5.3). Введем новые переменные

$$x_{ij} = \alpha_i h_{ij},$$

положим

$$b_j = \frac{s_j}{\beta_j}, \quad a_i = \alpha_i h_i, \quad c_{ij} = \frac{r_{ij}}{\alpha_i}.$$

Тогда, как нетрудно видеть, сформулированная задача переходит в следующую.

Требуется минимизировать линейную форму

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} \quad (5.12)$$

при соблюдении условий:

$$\sum_{j=1}^n x_{ij} \leq a_i, \quad i = 1, 2, \dots, m, \quad (5.13)$$

$$\sum_{i=1}^m x_{ij} = b_j, \quad j = 1, 2, \dots, n, \quad (5.14)$$

$$x_{ij} \geq 0. \quad (5.15)$$

Задача же (5.12), (5.13), (5.14), (5.15), как было показано в п. 1.2 настоящей главы, без труда сводится к обычной задаче транспортировки T . Заметим, что для разрешимости задачи рационального использования станков необходимо и достаточно выполнение неравенства

$$\sum_{j=1}^n \frac{s_j}{\beta_j} \leq \sum_{i=1}^m \alpha_i h_i. \quad (5.16)$$

В самом деле, условие разрешимости задачи (5.12), (5.13), (5.14), (5.15), как мы видели, эквивалентно соотношению

$$\sum_{j=1}^n b_j \leq \sum_{i=1}^m a_i,$$

из которого с очевидностью вытекает (5.16). Некоторые задачи приводят к схеме задачи транспортировки, в которой необходимо не минимизировать, а максимизировать форму (5.3). Примером такой задачи может служить рассматриваемая в § 3 этой главы проблема выбора. Если в подобной задаче изменить знак перед матрицей транспортных издержек, то она сведется к обычной проблеме транспортировки. Поэтому все результаты, установленные для задачи T , справедливы также и для задачи мак-

симизации транспортного типа. Несколько задач, сводящихся к транспортной, рассмотрено в § 10 гл. II.

Перейдем теперь к описанию методов решения транспортной проблемы.

§ 2. МЕТОД ПОТЕНЦИАЛОВ

2.1. Предварительные замечания. Как уже отмечалось, метод потенциалов был описан [11] раньше других методов решения транспортной задачи. Согласно методу потенциалов одновременно с получением оптимального плана транспортной задачи на множестве ее пунктов определяется некоторая функция, значения которой естественно называть потенциалами соответствующих пунктов. Этим и объясняется название метода. По существу, метод потенциалов является детализацией метода последовательного улучшения плана применительно к транспортной проблеме. Однако впервые он был изложен вне связи с общими методами линейного программирования*). Напомним вкратце общую схему метода последовательного улучшения плана. Вначале выбирается некоторый исходный план задачи. Затем этот план исследуется на оптимальность. В случае его неоптимальности указывается прием, с помощью которого осуществляется переход к новому плану, связанному с бóльшим значением линейной формы задачи (речь идет о задаче максимизации). Указанный процесс повторяется до получения решения исследуемой задачи. Метод последовательного улучшения плана может быть реализован как с помощью алгоритма I, так и с помощью алгоритма II. Поскольку при больших величинах n и m число условий в транспортной задаче $(n + m)$ существенно меньше числа ее переменных (nm) , при ее решении предпочтение следует оказать II алгоритму **). Именно этот алгоритм и используется в методе потенциалов. В § 3 гл. 6 мы подробнее остановимся на отмеченной связи между обоими методами. Теперь же приступим к описанию метода потенциалов и связанного с ним алгоритма.

*) К моменту опубликования работы [11] общие методы линейного программирования не были еще подробно развиты. Впрочем основные их идеи уже содержались в работе Л. В. Канторовича [6].

**) На целесообразность выбора II алгоритма указывает также большое число нулевых компонент в векторах условий транспортной задачи.

2.2. Описание метода потенциалов. При изложении метода часть необходимых утверждений будет лишь формулироваться. Что касается соответствующих доказательств, то они будут проведены в § 3 гл. 6. Вначале метод потенциалов будет рассмотрен применительно к невырожденной задаче. На изменениях, которые необходимо в него внести в случае вырожденности, мы остановимся в конце параграфа. Метод состоит из конечного числа итераций, с помощью которых по некоторому исходному плану задачи строится ее решение (оптимальный план). Каждая итерация разбивается на 2 этапа. На 1-м этапе план, полученный в результате предыдущей итерации, проверяется на оптимальность. Если план оказывается решением, процесс заканчивается. Если же это не так, осуществляется переход к этапу 2. На 2-м этапе строится новый план, приводящий к меньшим (по сравнению с предыдущим планом) транспортным издержкам. Перед тем, как начать детальное рассмотрение метода, введем несколько необходимых понятий.

Пусть $X = \|x_{ij}\|_{m,n}$ — произвольный план задачи T . Элемент матрицы транспортных издержек c_{ij} будем называть X — существенным, если $x_{ij} > 0$ (перевозка из A_i в B_j по плану X положительна).

Функцию W , определенную на совокупности пунктов производства и потребления задачи T , назовем потенциалом этой задачи, если

$$a) \quad W(B_j) - W(A_i) \leq c_{ij},$$

$$i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n; \quad (5.17)$$

$$б) \quad W(B_j) - W(A_i) = c_{ij} \quad (5.18)$$

для всех X — существенных элементов некоторого плана X .

Таким образом, понятие потенциала связывается пока с определенным планом задачи T . Будем называть план X потенциальным, если существует потенциал задачи T , связанный с этим планом условием (5.18). Между оптимальностью и потенциальностью планов задачи T существует тесная связь. В § 3 гл. 6 будет доказан следующий критерий оптимальности: для оптималь-

ности плана X необходима и достаточна его потенциальность.

Заметим, что потенциал вовсе не следует считать зависящим от конкретного оптимального плана задачи T (в силу сформулированного критерия оптимальности функция W может быть связана лишь с оптимальным планом этой задачи). В дальнейшем (§ 3 гл. 6) мы убедимся, что каждый потенциал задачи T связан условием (5.18) с любым ее оптимальным планом (естественно, что это утверждение имеет смысл лишь для транспортных задач с неединственным решением). Поэтому функция W может быть определена еще и так:

W — потенциал задачи T , если

$$W(B_j) - W(A_i) \leq c_{ij}, \quad i = 1, 2, \dots, m, j = 1, 2, \dots, n,$$

причем для тех c_{ij} , которые являются X -существенными элементами произвольного оптимального плана X этой задачи, соответствующие неравенства переходят в равенства. Итак, функция W определяется полной совокупностью оптимальных планов данной задачи.

Значения потенциала W в пунктах задачи T будем называть потенциалами этих пунктов. Функция W не случайно названа потенциалом задачи. Смысл этого названия можно иллюстрировать следующей аналогией.

Представим себе, что некую единичную массу необходимо перенести из точки A_i в точку B_j . Величина работы, которую необходимо совершить при переносе массы из A_k в B_l $k = 1, 2, \dots, m$; $l = 1, 2, \dots, n$, задана и равна c_{kl} (работа, совершаемая при движении от B_l к A_k , предполагается равной $-c_{kl}$). Условимся, что непосредственное движение от A_k к B_l (или наоборот) возможно лишь в том случае, если согласно некоторому оптимальному плану задачи T по этой коммуникации направляется положительная перевозка. Пусть движение от A_i к B_j запланировано следующим образом:

$$A_i \rightarrow B_{j_1} \rightarrow A_{i_1} \rightarrow B_{j_2} \rightarrow \dots \rightarrow A_{i_s} \rightarrow B_j.$$

Тогда производимая при этом работа вычисляется по формуле

$$A = c_{ij_1} - c_{i_1j_1} + c_{i_1j_2} - \dots - c_{i_sj_s} + c_{i_sj}.$$

Используя теперь свойства потенциала задачи T , имеем

$$\begin{aligned} A = [W(B_{j_1}) - W(A_{i_1})] - [W(B_{j_1}) - W(A_{i_1})] + \dots \\ \dots - [W(B_{j_s}) - W(A_{i_s})] + [W(B_j) - \\ - W(A_{i_s})] = W(B_j) - W(A_i). \end{aligned}$$

Итак, величина $W(B_j) - W(A_i)$ совпадает с количеством работы, необходимым для переноса единичной массы из A_i в B_j . При этом выполняется основное свойство движения в потенциальном поле: величина работы, производимой при перемещении, не зависит от формы возможного пути, связывающего рассматриваемые точки, а определяется лишь самими точками. Заметим, что в силу (5.17) непосредственное движение от A_k к B_l всегда связано с неменьшей работой по сравнению с движением между этими точками по любому из возможных маршрутов.

После этих предварительных замечаний перейдем к описанию метода потенциалов. Предположим, что уже проведено k итераций метода потенциалов, причем последняя привела к построению плана $X_k = \|x_{ij}^{(k)}\|_{m,n}$ (первоначальный план $X_0 = \|x_{ij}^{(0)}\|$ предполагается опорным). Разберем подробно следующую, $(k+1)$ -ю итерацию.

Этап 1. На этом этапе производится исследование плана X_k на оптимальность. Обозначим через G_k совокупность X_k — существенных элементов матрицы C , т. е. таких c_{ij} , которым соответствуют $x_{ij}^{(k)} > 0$, и займемся определением величин $u_i^{(k)}$, $i = 1, 2, \dots, m$; $v_j^{(k)}$, $j = 1, 2, \dots, n$, удовлетворяющих системе уравнений

$$v_j^{(k)} - u_i^{(k)} = c_{ij}, \text{ для всех } c_{ij} \in G_k. \quad (5.19)$$

В силу невырожденности исследуемой задачи любые два пункта A_i, B_j можно соединить цепочкой, состоящей из коммуникаций с ненулевыми (по плану X_k) перевозками. Далее, из опорности плана X_0 вытекает опорность плана X_k (гл. 6, § 3) и, следовательно, согласно теореме I предыдущего параграфа в нем нет замкнутых цепочек. Поэтому отмеченная цепочка строится однозначно. С учетом приведенных соображений решение системы (5.19) проводится чрезвычайно просто. Зададимся

значением одной из неизвестных системы (5.19), например, положим $u_1^{(k)} = 0$. Допустим, что пункт A_1 снабжает в соответствии с планом X_k пункты $B_{j_1}, B_{j_2}, \dots, B_{j_s}$. Тогда из соответствующих уравнений системы (5.19) определяем

$$v_{j_\lambda}^{(k)} = c_{1j_\lambda}, \quad \lambda = 1, 2, \dots, s.$$

Далее, аналогичным способом определим значения $u_i^{(k)}$ для тех A_i , которые снабжают один из пунктов B_{j_λ} , $\lambda = 1, 2, \dots, s$. Например, если A_{i_t} снабжает пункт B_{j_1} , то

$$u_{i_t}^{(k)} = v_{j_1}^{(k)} - c_{i_t j_1}.$$

Затем определяются $v_j^{(k)}$ для B_j , снабжающихся за счет пунктов A_i с уже вычисленными значениями $u_i^{(k)}$, и т. д. Таким образом, задавшись произвольным значением $u_1^{(k)}$, мы легко определим $u_i^{(k)}$, $v_j^{(k)}$ для всех тех пунктов A_i , B_j , которые можно соединить с A_1 цепочкой из коммуникаций с ненулевыми перевозками. Но, как было отмечено, подобной цепочкой могут быть соединены любые два пункта рассматриваемой задачи и притом единственным образом. Следовательно, при заданном $u_1^{(k)}$ величины $u_i^{(k)}$, $v_j^{(k)}$ определяются однозначно для всех пунктов задачи T . Нетрудно усмотреть, что если бы мы задались произвольным значением любой из величин $u_i^{(k)}$ или $v_j^{(k)}$, то получившиеся в результате решения системы (5.19) числа $\bar{u}_i^{(k)}$, $\bar{v}_j^{(k)}$ отличались бы от $u_i^{(k)}$, $v_j^{(k)}$, вычисленных в предположении $u_1^{(k)} = 0$, на некоторую постоянную. Следовательно, для любых i, j величина $v_j^{(k)} - u_i^{(k)}$ определяется однозначно.

Если величины $u_i^{(k)}$, $v_j^{(k)}$ таковы, что для любой пары i, j имеет место неравенство

$$v_j^{(k)} - u_i^{(k)} \leq c_{ij},$$

то функция W , определяемая условиями

$$W(A_i) = u_i^{(k)}, \quad W(B_j) = v_j^{(k)},$$

— потенциал задачи T , и, следовательно, X_k , будучи потенциальным, является, вместе с тем и оптимальным

планом исследуемой задачи. Если же для некоторых индексов i, j

$$v_j^{(k)} - u_i^{(k)} > c_{ij},$$

план X_k неоптимальный, а поэтому необходим переход к этапу 2, на котором этот план будет улучшен.

Этап 2. Вычислим уклонения

$$c_{ij}^{(k+1)} = c_{ij} - (v_j^{(k)} - u_i^{(k)}).$$

По условию среди чисел $c_{ij}^{(k+1)}$ имеются отрицательные. Определим i_0, j_0 из условия

$$c_{i_0 j_0}^{(k+1)} = \min_{i, j} c_{ij}^{(k+1)}$$

(в случае, если таких пар индексов несколько, выберем одну из них). Соединим пункты A_{i_0}, B_{j_0} цепочкой из коммуникаций, по которым согласно плану X_k намечены ненулевые перевозки (мы уже отмечали, что это возможно, и притом единственным способом). Пусть цепочка проходит последовательно через пункты

$$A_{i_0}, B_{j_1}, A_{i_2}, B_{j_2}, \dots, B_{j_{s-1}}, A_{i_s}, B_{j_0}.$$

Рассмотрим перевозки по тем коммуникациям построенной цепочки, которые при движении от A_{i_0} к B_{j_0} проходятся в положительном направлении, т. е. от пункта производства к пункту потребления. Минимальную среди них обозначим черед θ_k . Таким образом,

$$\theta_k = \min_{1 \leq \lambda \leq s} x_{i_\lambda j_\lambda}^{(k)} \quad (i_0 = i_1, j_0 = j_s).$$

Введем теперь в план X_k следующие изменения: величины перевозок от A_{i_λ} к B_{j_λ} , $\lambda = 1, 2, \dots, s$, уменьшим на θ_k , величины перевозок от $A_{i_{\lambda+1}}$ к B_{j_λ} , $\lambda = 1, \dots, s-1$, увеличим на θ_k (здесь, как и прежде, полагаем $i_0 = i_1, j_0 = j_s$). Кроме того, введем новую перевозку между пунктами A_{i_0}, B_{j_0} , по величине равную θ_k . Очевидно, что новая совокупность перевозок является планом исследуемой транспортной задачи. Действительно, все перевозки этой совокупности в силу определения θ_k неотрицательны. Далее, для любого пункта, входящего

в цепочку, количество вывозимого (для пунктов производства) и ввозимого (для пунктов потребления) продукта не меняется. Для того чтобы убедиться в этом, рассмотрим один из пунктов A_{i_λ} , $\lambda = 1, 2, \dots, s$. В соответствии с новой совокупностью перевозок количество продукта, перевозимого из A_{i_λ} в B_{j_λ} , уменьшается на θ_k и одновременно на эту же величину увеличивается грузопоток из A_{i_λ} в $B_{j_{\lambda-1}}$.

Подобные же рассуждения показывают, что общее количество продукта, ввозимого в любой из пунктов B_{j_λ} , также не меняется. Что касается пунктов, не входящих в цепочку, то условия транспортной задачи для них, естественно, не нарушатся, т. к. ни одна из перевозок, связывающих эти пункты, не изменилась. Заметим, что по крайней мере одна из перевозок цепочки в силу определения θ_k обратится в нуль. Из предположения о невырожденности задачи следует, что более одной перевозки в нуль обратиться не может. Поскольку величина перевозки из A_{i_0} в B_{j_0} стала теперь положительной, новый план, который мы обозначим через X_{k+1} , по-прежнему состоит из $n+m-1$ перевозок. Покажем, что реализация плана X_{k+1} связана с меньшими транспортными издержками по сравнению с планом X_k . Действительно,

$$L_{k+1} = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^{(k+1)} = \sum_I c_{ij} x_{ij}^{(k)} + \\ + \sum_{\lambda=1}^s c_{i_\lambda j_\lambda} (x_{i_\lambda j_\lambda}^{(k)} - \theta_k) + \sum_{\lambda=1}^{s-1} c_{i_{\lambda+1} j_\lambda} (x_{i_{\lambda+1} j_\lambda}^{(k)} + \theta_k) + c_{i_0 j_0} \theta_k,$$

где под $\sum_I$ понимается суммирование по перевозкам, не входящим в цепочку; i_0, j_0 равны соответственно i_1, j_s . Совершая очевидные преобразования, получаем

$$L_{k+1} = L_k + \theta_k [c_{i_0 j_0} - c_{i_0 j_1} + c_{i_2 j_1} - \dots + c_{i_s j_{s-1}} - c_{i_s j_0}].$$

Вспомним, что все элементы, стоящие в квадратной скобке, кроме $c_{i_0 j_0}$, — X_k , существенны. Откуда

$$c_{i_\lambda j_\lambda} = v_{i_\lambda}^{(k)} - u_{j_\lambda}^{(k)}; \quad c_{i_{\lambda+1} j_\lambda} = v_{i_{\lambda+1}}^{(k)} - u_{j_\lambda}^{(k)}.$$

Учитывая далее отрицательность величины $c_{i_0 j_0}^{(k+1)}$, имеем

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^{(k+1)} &= L_{k+1} = L_k + \theta_k [c_{i_0 j_0} - \\ &- (v_{j_1}^{(k)} - u_{i_0}^{(k)}) + \dots + (v_{j_{s-1}}^{(k)} - u_{i_s}^{(k)}) - (v_{j_0}^{(k)} - u_{i_0}^{(k)})] = \\ &= L_k + \theta_k [c_{i_0 j_0} - (v_{j_0}^{(k)} - u_{i_0}^{(k)})] = \\ &= L_k + \theta_k c_{i_0 j_0}^{(k+1)} < L_k. \end{aligned} \quad (5.20)$$

Итак, составлен новый план перевозок X_{k+1} , соответствующий меньшему значению линейной формы транспортной задачи. Этап 2, а вместе с ним и $(k+1)$ -я итерация закончены. В обосновании метода (гл. 6, § 3) будет показано, что через конечное число итераций мы обязательно придем к искомому решению рассматриваемой задачи транспортировки.

Все вычисления, связанные с использованием метода потенциалов, можно свести в достаточно компактную схему, составляющую алгоритм метода.

2.3. Описание алгоритма. Алгоритм складывается из предварительного этапа и конечного числа последовательно проводимых итераций. В каждой итерации рассматриваются и преобразовываются две матрицы $\|x_{ij}^{(k)}\|_{m,n}$; $\|c_{ij}^{(k)}\|_{m,n}$.

Предварительный этап.

а) Используя метод северо-западного угла или разбираемый в следующем пункте метод минимального элемента, определим некоторый опорный план X_0 исследуемой задачи.

б) По правилам, изложенным при описании 1-го этапа метода потенциалов, вычисляем величины $u_i^{(0)}$, $v_j^{(0)}$, $i=1, 2, \dots, m$, $j=1, 2, \dots, n$ и составляем матрицу

$$C_1 = \|c_{ij} - (v_j^{(0)} - u_i^{(0)})\|_{m,n}.$$

Каждая итерация алгоритма состоит из двух этапов *. Как и при описании метода, предположим, что прове-

* Это относится ко всем итерациям, кроме первой, которая содержит лишь этап 2, и последней, состоящей из этапа 1.

дено k итераций, в результате которых получен план X_k , а также матрица

$$C_k = \|c_{ij} - (v_j^{(k-1)} - u_i^{(k-1)})\|_{m,n}.$$

Целью $(k+1)$ -й итерации является либо установление оптимальности плана X_k , либо построение нового, более экономичного плана X_{k+1} и связанной с ним матрицы C_{k+1} . Рассмотрим оба этапа $(k+1)$ -й итерации.

Этап 1. Выбираем отрицательный X_k — существенный элемент матрицы $C_k : c_{\lambda\mu}^{(k)} = \delta_{k+1}$ (такой элемент обязательно существует и единственен, см. конец этапа 2) *). Выделим содержащую его строку и через G_1 обозначим совокупность X_k — существенных элементов этой строки, не совпадающих с δ_{k+1} . Затем выделим столбцы C_k , содержащие элементы G_1 , и через G_2 обозначим множество X_k — существенных элементов, лежащих в этих столбцах и отличных от элементов G_1 . Далее выделяются строки C_k , содержащие элементы G_2 , и аналогично предыдущему вводится множество G_3 . Процесс выделения строк и столбцов матрицы C_k продолжается до тех пор, пока очередное множество G_l , скажем G_l , не окажется пустым. Заметим, что поскольку каждая строка (каждый столбец) не может быть выделена (выделен) более одного раза (X_k — опорный план!), $l \leq n + m - 1$. Закончив выделение линий (строк или столбцов) матрицы C_k , приступим к построению матрицы C_{k+1} . Для этого величину δ_{k+1} прибавим ко всем выделенным столбцам и вычтем из всех выделенных строк матрицы C_k . Очевидно, что матрица, полученная после указанных преобразований, имеет вид

$$\|c_{ij} + \lambda_i - \mu_j\|_{m,n}.$$

Кроме того, все X_k — существенные ее элементы — нули. Действительно, после вычитания δ_{k+1} из λ -й строки C_k элемент $c_{\lambda\mu}^{(k)}$ обращается в нуль, а все элементы G_1 становятся равными $-\delta_{k+1}$. После прибавления δ_{k+1} к столбцам элементов G_1 эти элементы обращаются в нуль, а элементы G_2 принимают значение δ_{k+1} .

*) Этот элемент совпадает с Δ_k .

Заметим, что поскольку линии C_k отмечаются не более одного раза, элементы G_1 при дальнейших преобразованиях остаются равными нулю. Продолжая вычитать (из выделенных строк) и прибавлять (к выделенным столбцам) матрицы C_k величину δ_{k+1} , обращаем последовательно в нуль элементы G_2, G_3 и т. д.

После $l-1$ шагов все X_k — существенные элементы C_k , кроме принадлежащих G_{l-1} , оказываются равными нулю. Пусть l — нечетное (четное) число. Так как G_l — пустое множество, после вычитания (прибавления) величины δ_{k+1} из строк (к столбцам) элементов G_{l-1} все X_k — существенные элементы полученной матрицы — нули. Итак, величины $\lambda_i, \mu_j, i=1, 2, \dots, m; j=1, 2, \dots, n$ удовлетворяют системе (5.19), соответствующей плану X_k . Следовательно,

$$\begin{aligned} \mu_j - \lambda_i &= v_j^{(k)} - u_i^{(k)}, \quad C_{k+1} = \|c_{ij} + \lambda_i - \mu_j\|_{m,n} = \\ &= \|c_{ij} - (v_j^{(k)} - u_i^{(k)})\|_{m,n}. \end{aligned}$$

Если все элементы матрицы C_{k+1} неотрицательны, X_k — искомое решение. Если же среди величин $c_{ij}^{(k+1)}$ имеются отрицательные, переходим к этапу 2.

Этап 2. Выберем наименьший элемент матрицы C_{k+1} , расположенный, скажем, на пересечении i_0 -й строки и j_0 -го столбца, и обозначим его через Δ_{k+1} . По условию $\Delta_{k+1} < 0$. Переходим к поиску цепочки из положительных элементов матрицы X_k , замыкающейся на элементе $x_{i_0 j_0}^{(k)}$ (очевидно, $x_{i_0 j_0}^{(k)} = 0$). Соединим стрелками $x_{i_0 j_0}$ со всеми положительными элементами его строки, совокупность которых обозначим через $\bar{G}_1$. Затем каждый из элементов $\bar{G}_1$ соединим стрелками со всеми положительными элементами, расположенными в его столбце. Совокупность всех таких элементов обозначим через $\bar{G}_2$. Процесс образования множеств $\bar{G}_k$ продолжается до тех пор, пока при некотором p (p — нечетное число) среди столбцов, содержащих элементы $\bar{G}_p$, не обнаружится столбец с номером j_0 . Заметим, что в силу опорности плана X_k множества $\bar{G}_\lambda, \bar{G}_\mu, \lambda \neq \mu$ не пересекаются, ибо в противном случае существовала бы замкнутая цепочка из ненулевых (согласно X_k) перевозок. Это обстоятельство, а также то, что любые

два пункта невырожденной задачи транспортировки могут быть соединены коммуникациями с ненулевыми (по плану X_k) перевозками, служит доказательством существования введенного выше числа $p \leq n + m - 1$. Теперь уже не трудно образовать искомую цепочку. От элемента $x_{i_0 j_0}^{(k)}$ перейдем по столбцу к элементу $x_{i_s j_0}^{(k)} \in \bar{G}_p$. От $x_{i_s j_0}^{(k)}$ по строке перейдем к соединенному с ним стрелкой элементу $x_{i_s j_s}^{(k)} \in \bar{G}_{p-1}$ и т. д. Двигаясь таким образом вдоль намеченных стрелок по строкам и столбцам матрицы X_k , получим в конце концов последовательность положительных элементов этой матрицы вида:

$$x_{i_0 j_1}^{(k)}, x_{i_1 j_1}^{(k)}, \dots, x_{i_s j_s}^{(k)}, x_{i_s j_0}^{(k)} (2s + 1 = p),$$

образующую цепочку, замыкающуюся на элементе $x_{i_0 j_0}^{(k)}$. Обозначим совокупность пар индексов $(i_\mu, j_{\mu+1})$, $\mu = 0, 1, \dots, s$, ($j_{s+1} = j_0$) через I_1 , совокупность пар индексов вида (i_μ, j_μ) , $\mu = 1, 2, \dots, s$, — через I_2 , объединение I_1, I_2 (множество пар, принадлежащих либо I_1 , либо I_2) через I . Положим

$$\theta_{k+1} = \min_{0 \leq \mu \leq s} x_{i_\mu j_{\mu+1}}^{(k)}. \quad (5.20)$$

Новый план X_{k+1} определяется согласно правилу

$$x_{ij}^{(k+1)} = \begin{cases} \theta_{k+1}, & i, j = i_0 j_0, \\ x_{ij}^{(k)} - \theta_{k+1}, & i, j \in I_1, \\ x_{ij}^{(k)} + \theta_{k+1}, & i, j \in I_2, \\ x_{ij}^{(k)}, & i, j \notin I. \end{cases} \quad (5.21)$$

(Из нечетных элементов цепочки θ_{k+1} вычитается, к четным элементам и $x_{i_0 j_0}^{(k)} \theta_{k+1}$ прибавляется, остальные элементы плана X_k не изменяются). Итак, в результате $(k+1)$ -й итерации от плана X_k и связанной с ним матрицы $C_k = \|c_{ij} - (v_j^{(k-1)} - u_i^{(k-1)})\|_{m,n}$ мы перешли к улучшенному плану X_{k+1} и связанной с ним мат-

рице $C_{k+1} = \|c_{ij} - (v_j^{(k)} - u_i^{(k)})\|_{m,n}$. Из способа построения плана X_{k+1} следует, что среди X_{k+1} — существенных элементов C_{k+1} лишь один $(c_{l_0 j_0}^{(k+1)})$ не равен нулю.

Пример 2.1. Решить транспортную задачу, определяемую таблицей:

		Пункты потребления				
		1	2	3	4	
Пункты производства	1	7	8	5	3	11
	2	2	4	5	9	11
	3	6	3	1	2	8
		5	9	9	7	
		Объемы потребления				

Анализируя объемы производства и потребления различных пунктов задачи, убеждаемся в ее невырожденности. Поэтому для решения можно использовать рассмотренный алгоритм метода потенциалов. Прежде чем приступить к выкладкам, сделаем несколько замечаний. Для наглядности X_k — существенные элементы матрицы C_k будем отмечать черточками. При вычислении C_k удобно справа от выделенных строк матрицы C_{k-1} помещать $-\delta_k$, под выделенными столбцами ставить $+\delta_k$. Процесс выделения строк и столбцов отмечается стрелками.

Минимальный элемент матрицы C_k Δ_k , который определяет начало цепочки в X_k , будем обводить кружком. Элементы цепочки в X_k отмечаются звездочками. После каждой итерации выписывается соответствующее значение линейной формы задачи (транспортных издержек) L_k . Заметим, что для контроля полезно вычислять L_k двумя способами: непосредственно и по рекуррентной формуле

$$L_k = L_{k-1} + \theta_k \Delta_k.$$

Предварительный этап. а) Определяем методом северо-западного угла исходный опорный план рассматриваемой задачи

5	6	0	0		11	6	0	0	0	0
0	3	8	0		11	11	11	8	0	0
0	0	1	7		8	8	8	8	8	0

5	9	9	7							
0	9	9	7							
0	3	9	7							
0	0	9	7							
0	0	1	7							
0	0	0	0							

$$; X_0 = \begin{vmatrix} 5 & 6 & 0 & 0 \\ 0 & 3 & 8 & 0 \\ 0 & 0 & 1 & 7 \end{vmatrix}.$$

Соответствующие X_0 суммарные транспортные издержки составляют $L_0 = 150$ ед.

б) Строим матрицу

$$C_1 = \| c_{ij} - (v_j^{(0)} - u_i^{(0)}) \|_{3,4}.$$

Для этого находим величины:

$$u_i^{(0)}, i = 1, 2, 3; \quad v_j^{(0)}, j = 1, 2, 3, 4,$$

решая систему (5.19) для плана X_0 :

$$v_1^{(0)} - u_1^{(0)} = 7, \quad v_2^{(0)} - u_1^{(0)} = 8,$$

$$v_2^{(0)} - u_2^{(0)} = 4, \quad v_3^{(0)} - u_2^{(0)} = 5,$$

$$v_3^{(0)} - u_3^{(0)} = 1, \quad v_4^{(0)} - u_3^{(0)} = 2.$$

Полагаем $u_1^{(0)} = 0$. Откуда

$$u_2^{(0)} = 4; \quad u_3^{(0)} = 8; \quad v_1^{(0)} = 7; \quad v_2^{(0)} = 8;$$

$$v_3^{(0)} = 9; \quad v_4^{(0)} = 10.$$

$$C_1 = \begin{vmatrix} 0 & 0 & -4 & \boxed{-7} \\ -1 & 0 & 0 & 3 \\ 7 & 3 & 0 & 0 \end{vmatrix}.$$

Итерация I.

Этап 1. Отсутствует, так как C_1 уже определена.

Этап 2.

$$X_0 = \begin{vmatrix} 5 & 6^* & 0 & 0 \\ 0 & 3^* & 8^* & 0 \\ 0 & 0 & 1^* & 7^* \end{vmatrix} \xrightarrow{\theta_1=6} X_1 = \begin{vmatrix} 5 & 0 & 0 & 6 \\ 0 & 9 & 2 & 0 \\ 0 & 0 & 7 & 1 \end{vmatrix}.$$

$L_1 = 108.$

Итерация II.

Этап 1.

$$C_1 = \begin{vmatrix} \bar{0} & 0 & -4 & -7 \\ -1 & \bar{0} & \bar{0} & 3 \\ 7 & 3 & \bar{0} & \bar{0} \\ -7 & & & \end{vmatrix} \xrightarrow{+7} C_2 = \begin{vmatrix} 0 & 7 & 3 & 0 \\ \boxed{-8} & 0 & 0 & 3 \\ 0 & 3 & 0 & 0 \end{vmatrix}.$$

Этап 2.

$$X_1 = \begin{vmatrix} 5^* & 0 & 0 & 6^* \\ 0 & 9 & 2^* & 0 \\ 0 & 0 & 7^* & 1^* \end{vmatrix} \xrightarrow{\theta_2=1} X_2 = \begin{vmatrix} 4 & 0 & 0 & 7 \\ 1 & 9 & 1 & 0 \\ 0 & 0 & 8 & 0 \end{vmatrix}.$$

$L_2 = 100.$

Итерация III.

Этап 1.

$$C_2 = \begin{vmatrix} \bar{0} & 7 & 3 & \bar{0} \\ -\bar{8} & \bar{0} & \bar{0} & 3 \\ 0 & 3 & \bar{0} & 0 \\ -8 & -8 & & \end{vmatrix} \xrightarrow{+8} C_3 = \begin{vmatrix} 0 & -1 & \boxed{-5} & 0 \\ 0 & 0 & 0 & 11 \\ 8 & 3 & 0 & 8 \end{vmatrix}.$$

Этап 2.

$$X_2 = \begin{vmatrix} 4^* & 0 & 0 & 7 \\ 1^* & 9 & 1^* & 0 \\ 0 & 0 & 8 & 0 \end{vmatrix} \xrightarrow{\theta_3=1} X_3 = \begin{vmatrix} 3 & 0 & 1 & 7 \\ 2 & 9 & 0 & 0 \\ 0 & 0 & 8 & 0 \end{vmatrix}.$$

$L_3 = 95.$

Итерация IV.

Этап 1.

$$C_3 = \begin{vmatrix} \bar{0} & -1 & -5 & \bar{0} \\ \bar{0} & \bar{0} & 0 & 11 \\ 8 & 3 & \bar{0} & 8 \\ -5 & -5 & & -5 \end{vmatrix} \xrightarrow{\substack{+5 \\ +5}} C_4 = \begin{vmatrix} 0 & -1 & 0 & 0 \\ 0 & 0 & 5 & 11 \\ 3 & \boxed{-2} & 0 & 3 \end{vmatrix}.$$

Этап 2.

$$X_3 = \begin{vmatrix} 3^* & 0 & 1^* & 7 \\ 2^* & 9^* & 0 & 0 \\ 0 & 0 & 8^* & 0 \end{vmatrix} \xrightarrow[\theta_4 = -3]{L = 89.} X_4 = \begin{vmatrix} 0 & 0 & 4 & 7 \\ 5 & 6 & 0 & 0 \\ 0 & 3 & 5 & 0 \end{vmatrix}.$$

Итерация V.

Этап 1.

$$C_4 + \begin{vmatrix} 0 & -1 & \bar{0} & \bar{0} \\ \bar{0} & \bar{0} & 5 & 11 \\ 3 & -2 & \bar{0} & 3 \\ & -2 & -2 & \end{vmatrix} \xrightarrow{+2} C_5 = \begin{vmatrix} 2 & 1 & 0 & 0 \\ 0 & 0 & 3 & 9 \\ 5 & 0 & 0 & 3 \end{vmatrix}.$$

Все элементы C_5 неотрицательны. Следовательно,

$$X_4 = \begin{vmatrix} 0 & 0 & 4 & 7 \\ 5 & 6 & 0 & 0 \\ 0 & 3 & 5 & 0 \end{vmatrix} \text{ — искомое решение задачи (оптимальный план).}$$

Поскольку среди элементов матрицы C_1 имеются отрицательные, план X_0 не является оптимальным и необходим переход к итерации I. Итерация I содержит лишь этап 2 (первый этап, состоящий в построении матрицы C_1 , уже проведен).

Выбираем минимальный элемент матрицы C_1 $c_{14}^{(1)} = -7 = \Delta_1$ и приступаем к построению цепочки из положительных элементов матрицы X_0 , замыкающейся

на $x_{14}^{(0)}$. В соответствии с алгоритмом соединяем стрелками $x_{14}^{(0)}$ с положительными элементами первой строки матрицы X_0 — $x_{11}^{(0)}$, $x_{12}^{(0)}$ (они составляют множество $\bar{G}_1$). В столбцах, исходящих из элементов $\bar{G}_1$ (столбцы 1, 2), имеется всего один положительный элемент, не принадлежащий $\bar{G}_1$ ($x_{22}^{(0)}$). Этот элемент, составляющий множество $\bar{G}_2$, соединяется стрелкой с $x_{12}^{(0)} \in \bar{G}_1$. Каждое из последующих множеств $\bar{G}_3$, $\bar{G}_4$, $\bar{G}_5$, как нетрудно заметить, содержит по одному элементу. Поскольку единственный элемент множества $\bar{G}_5$ ($x_{34}^{(0)}$) расположен в одном столбце с $x_{14}^{(0)}$, процесс образования множеств $\bar{G}_i$ прекращается. Теперь нетрудно построить искомую цепочку: движемся от $x_{34}^{(0)}$ по строке до соединенного с ним (стрелкой) элемента $x_{33}^{(0)}$. Затем от $x_{33}^{(0)}$ переходим по столбцу к $x_{23}^{(0)}$. Далее, включаем в цепочку элементы $x_{22}^{(0)}$, $x_{12}^{(0)}$. Итак, цепочка матрицы X_0 составляется из элементов $x_{12}^{(0)}$, $x_{22}^{(0)}$, $x_{23}^{(0)}$, $x_{33}^{(0)}$, $x_{34}^{(0)}$ (в матрице X_0 они отмечаются звездочками). Переходим к построению нового плана X_1 . Для этого из нечетных элементов, составляющих цепочку, выбираем минимальный $\theta_1 = \min\{6, 8, 7\} = 6$. Далее, к четным элементам цепочки ($x_{22}^{(0)}$, $x_{33}^{(0)}$) и элементу $x_{14}^{(0)}$, на котором она замыкается, прибавляем 6; из нечетных элементов цепочки 6 вычитаем. Получаем новый план X_1 , соответствующий транспортным издержкам $L_1 = 108$. Величину L_1 можно подсчитать двумя способами

$$L_1 = \sum_{i=1}^3 \sum_{j=1}^4 c_{ij} x_{ij}^{(1)};$$

$$L_1 := L_0 + \theta_1 \Delta_1.$$

Итерация I закончена.

Итерация II содержит как этап 1, так и этап 2. На первом этапе план X_1 проверяется на оптимальность. Для этого матрица C_1 преобразуется в матрицу C_2 . Прежде всего в матрице C_1 отмечаются (черточками) X_1 — существенные элементы. Единственным ненулевым X_1 — существенным элементом C_1 является $c_{14}^{(1)} = \delta_2 = -7$. Далее, в соответствии с алгоритмом переходим к по-

строению множеств G_i , которые отмечаются стрелками в матрице C_1 . Множество G_1 состоит из одного элемента $c_{11}^{(1)}$. Поскольку в столбце, содержащем $c_{11}^{(1)}$ (первый столбец матрицы C_1), нет X_1 — существенных элементов, отличных от $c_{11}^{(1)}$, множество G_2 не содержит ни одного элемента; процесс образования множеств G_i прерывается. Выделяем строку матрицы C_1 , содержащую $c_{14}^{(1)}$ (строка 1), и столбец, в котором расположен единственный элемент G_1 (столбец 1). Вычитая из первой строки матрицы C_1 $\delta_2 = -7$ и прибавляя δ_2 к первому столбцу этой матрицы, получаем матрицу C_2 . Поскольку не все элементы матрицы C_2 неотрицательны ($c_{21}^{(1)} = -8$), план X_1 не является решением задачи. На втором этапе итерации II строится план X_2 , более экономичный по сравнению с X_1 . Для этого из положительных элементов X_1 составляется цепочка, замыкающаяся на $x_{21}^{(1)}$ ($c_{21}^{(2)} = \Delta_2 = -8$ — минимальный элемент матрицы C_2):

$$x_{23}^{(1)}, x_{33}^{(1)}, x_{34}^{(1)}, x_{14}^{(1)}, x_{11}^{(1)},$$

$$\theta_2 = \min \{x_{23}^{(1)}, x_{34}^{(1)}, x_{11}^{(1)}\} = \min \{2, 1, 5\} = 1.$$

Поэтому для получения плана X_2 необходимо из нечетных элементов цепочки вычесть 1, к четным ее элементам и элементу $x_{21}^{(1)}$ прибавить 1. Новый план соответствует транспортным издержкам

$$L_2 = 100 = L_1 + \theta_2 \Delta_2 = 108 - 8 = 100.$$

Следующие две итерации также содержат по два этапа.

Завершающей является итерация V. Она содержит лишь этап 1, на котором устанавливается оптимальность плана X_4 , построенного на втором этапе итерации IV.

2.4. Метод минимального элемента. Число итераций, необходимых для решения транспортной проблемы методом потенциалов, существенно зависит от вида первоначального плана. Удачный выбор исходного плана (первого приближения) может значительно сократить количество итераций и тем самым ускорить решение задачи. Поэтому важно иметь достаточно простой метод, позволяющий строить план, в большинстве случаев близкий к оптимальному. В предыдущем параграфе был описан один из методов построения плана задачи — метод северо-западного угла. Однако план, определяемый с помощью рассмотренного метода, зависит лишь от

объемов производства и потребления пунктов задачи и абсолютно не связан с матрицей транспортных издержек, вследствие чего этот план обычно значительно уклоняется от оптимального.

Вместе с тем ценой небольших усложнений можно модифицировать метод северо-западного угла, увязав естественным образом алгоритм построения плана со стоимостью перевозок. Эту модификацию обычно называют методом минимального элемента (смысл названия станет ясен из приводимого ниже описания метода).

Метод минимального элемента. Метод состоит из последовательно проводимых шагов.

Шаг 1. Определяем минимальный элемент матрицы транспортных издержек $C: c_{i_1 j_1}$ и полагаем

$$x_{i_1 j_1} = \min(a_{i_1}, b_{j_1}).$$

Возможны три случая:

$$1. \hat{a}_{i_1} < b_{j_1}; \quad 2. a_{i_1} > b_{j_1}; \quad 3. a_{i_1} = b_{j_1}.$$

В первом случае определяем элементы i_1 -й строки матрицы $X = \|x_{ij}\|_{m,n}$ (плана), полагая $x_{i,j} = 0, j \neq j_1$. Во втором — элементы j_1 -го столбца этой матрицы: $x_{i,j_1} = 0, i \neq i_1$. В последнем случае определяем и i_1 -ю строку и j_1 -й столбец X :

$$x_{i,j} = 0, \quad j \neq j_1; \quad x_{i,j_1} = 0, \quad i \neq i_1.$$

Шаг 2. Пусть C_1 — матрица, полученная из C вычеркиванием либо i_1 -й строки (случай 1), либо j_1 -го столбца (случай 2), либо и того и другого (случай 3). Положим

$$a_i^{(1)} = \begin{cases} a_i, & i \neq i_1 \\ a_i - x_{i,j_1}, & i = i_1 \end{cases}; \quad b_j^{(1)} = \begin{cases} b_j, & j \neq j_1 \\ b_j - x_{i_1,j}, & j = j_1 \end{cases}.$$

Очевидно, что общее число строк и столбцов C_1 по крайней мере на единицу меньше числа строк и столбцов матрицы C . Второй шаг состоит в проведении уже описанных операций применительно к матрице C_1 и величинам $a_i^{(1)}, b_j^{(1)}$. В результате этого шага заполняется по крайней мере одна линия (строка или столбец) матрицы X . Затем следует шаг 3 и т. д.

Процесс продолжается до полного заполнения матрицы $X = \|x_{ij}\|_{m,n}$. Особого обоснования предлагаемый метод не требует, так как легко редуцируется к рассмо-

тренному в § 1 методу северо-западного угла. Действительно, перенумеруем пункты производства и потребления задачи транспортировки в соответствии с порядком вычеркивания линий из матрицы C . Тогда применение к измененной задаче метода северо-западного угла приводит к плану $\tilde{X}$, совпадающему (после соответствующей перенумерации строк и столбцов) с матрицей, полученной методом минимального элемента.

Итак, метод минимального элемента всегда приводит к опорному плану исследуемой задачи транспортировки. Число шагов, необходимых для построения плана транспортной задачи с m пунктами производства и n пунктами потребления, не превосходит $n + m - 1$. Заметим, что каждый последующий шаг связан с меньшим количеством операций по сравнению с предыдущим.

Пример 2.2. Применим метод минимального элемента к транспортной задаче, решенной в примере 2.1.

$$C = \begin{vmatrix} 7 & 8^{(6)} & 5^{(5)} & 3^{(3)} \\ 2^{(2)} & 4^{(4)} & 5 & 9 \\ 6 & 3 & 1^{(1)} & 2 \end{vmatrix};$$

$$X = \begin{array}{cccc|cccccc} 0 & 3 & 1 & 7 & 11 & 11 & 11 & 4 & 4 & 0 \\ 5 & 6 & 0 & 0 & 11 & 11 & 6 & 6 & 0 & 0 \\ 0 & 0 & 8 & 0 & 8 & 0 & 0 & 0 & 0 & 0 \\ \hline 5 & 9 & 9 & 7 & \text{---} & & & & & \\ 5 & 9 & 1 & 7 & \text{---} & & & & & \\ 0 & 9 & 1 & 7 & \text{---} & & & & & \\ 0 & 9 & 1 & 0 & \text{---} & & & & & \\ 0 & 3 & 1 & 0 & \text{---} & & & & & \\ 0 & 0 & 0 & 0 & \text{---} & & & & & \end{array}$$

Итак

$$X = \begin{vmatrix} 0 & 3 & 1 & 7 \\ 5 & 6 & 0 & 0 \\ 0 & 0 & 8 & 0 \end{vmatrix}.$$

Для наглядности элемент матрицы C , оказавшийся минимальным на k -м шаге, снабжен сверху индексом

(k). Транспортные издержки построенного плана, как легко подсчитать, составляют 92 единицы и отличаются от минимальных всего на 3%. Применение к этой задаче метода северо-западного угла приводит, как мы видели, к плану с суммарными транспортными издержками в 150 единиц, отличающихся от минимальных на 68%.

Если в качестве первого приближения в задаче примера 2.1 взять только что построенный план, то ее решение уложится в неполные 2 итерации (вместо 5 итераций, имевших место ранее). Расчеты показывают, что в большинстве случаев метод минимального элемента позволяет получить достаточно хорошее первое приближение. Однако это вовсе не означает, что описанный метод во всех случаях дает лучшие результаты по сравнению с методом северо-западного угла. Приведем соответствующий пример. Рассмотрим транспортную задачу, определяемую таблицей

	3	5	7	8	10
	2	4	5	9	11
	6	3	1	2	8
	5	8	9	7	

(*)

План задачи (*), построенный методом минимального элемента, имеет вид

$$X_1 = \begin{vmatrix} 0 & 2 & 1 & 7 \\ 5 & 6 & 0 & 0 \\ 0 & 0 & 8 & 0 \end{vmatrix}$$

и приводит к транспортным издержкам $L_1 = 115$ ед. Применяя же к этой задаче метод северо-западного угла, приходим к плану

$$X_2 = \begin{vmatrix} 5 & 5 & 0 & 0 \\ 0 & 3 & 8 & 0 \\ 0 & 0 & 1 & 7 \end{vmatrix},$$

связанному с транспортными издержками $L_2 = 107$ ед. Таким образом, для рассматриваемой транспортной задачи план, построенный по методу северо-западного угла, оказался предпочтительней плана, определенного

методом минимального элемента. Как нетрудно проверить, план X_2 является решением задачи (*). Таким образом, хотя метод минимального элемента приводит в данном случае к худшему по сравнению с X_2 плану, связанные с ним транспортные издержки отличаются от минимальных всего на 7,5%.

2.5. Распространение метода на вырожденные задачи. Все рассуждения этого параграфа проводились до сих пор в предположении о невырожденности задачи T . Допустим теперь, что условия, гарантирующие согласно п. 1.4 невырожденность задачи T , не выполняются. Тогда исходный опорный план, построенный по одному из рассмотренных методов, может содержать меньше чем $n + m - 1$ положительных перевозок. Поэтому в его базис следует включить не только векторы коммуникации, соответствующие ненулевым перевозкам, но и несколько векторов с нулевыми перевозками. Это обстоятельство может, в свою очередь, привести к тому, что при переходе по методу потенциалов к следующему плану величина θ окажется равной 0 и, следовательно, суммарные транспортные издержки не уменьшатся. При построении базиса нового плана нам, как и в случае общей задачи линейного программирования, необходимо знать ответ на 2 вопроса: какой вектор следует ввести в старый базис, какой вектор (один!) необходимо из него вывести. При ответе на первый вопрос случай вырожденности не приводит ни к каким дополнительным трудностям. Что касается второго вопроса, то критерий, который использовался для ответа на него в невырожденном случае, в вырожденном случае оказывается недостаточным: нулевыми могут стать сразу несколько нечетных перевозок, составляющих цепочку этапа 2.

Можно было бы, конечно, выводить из старого базиса произвольный вектор коммуникаций из числа удовлетворяющих указанному критерию. Однако ввиду того, что в вырожденном случае суммарные транспортные издержки могут оставаться неизменными для ряда промежуточных планов, мы не можем гарантировать различие их базисов, а следовательно, ограниченность числа итераций метода потенциалов. При произвольном выводе одного из возможных векторов не исключен возврат через несколько итераций к старому базису, т. е. цикл.

Таким образом, необходимо иметь такие дополнительные правила исключения векторов из старых базисов, которые во всех случаях застраховали бы нас от зацикливания.

Пусть T — произвольная задача транспортировки. Свяжем с ней семейство транспортных задач $T(\epsilon)$, $\epsilon > 0$, объемы производства и потребления которых выражаются через соответствующие величины задачи T по формулам

$$a_i(\epsilon) = a_i + \epsilon;$$

$$b_j(\epsilon) = \begin{cases} b_j, & j \neq n, \\ b_j + m\epsilon, & j = n. \end{cases}$$

Покажем, что при достаточно малом ϵ ($0 < \epsilon < \delta$) $T(\epsilon)$ — невырожденная транспортная задача. Произвольная частная сумма объемов производства $a_i(\epsilon)$ имеет вид

$$\sum_{k=1}^t a_{i_k}(\epsilon) = \sum_{k=1}^t a_{i_k} + t\epsilon, \quad 0 < t < m.$$

Произвольная частная сумма объемов потребления

$$\sum_{k=1}^s b_{j_k}(\epsilon) = \begin{cases} \sum_{k=1}^s b_{j_k} \\ \sum_{k=1}^s b_{j_k} + m\epsilon \end{cases}, \quad 0 < s < n.$$

В силу критерия невырожденности транспортной задачи для доказательства нашего утверждения достаточно проверить, что

$$\sum_{k=1}^t a_{i_k}(\epsilon) \neq \sum_{k=1}^s b_{j_k}(\epsilon).$$

Рассуждая от противного, положим, что эти частные суммы совпадают. Тогда

$$\sum_{k=1}^t a_{i_k} - \sum_{k=1}^s b_{j_k} = \begin{cases} (m-t)\epsilon \\ -t\epsilon. \end{cases} \quad (5.22)$$

Возможны два случая:

$$1. \sum_{k=1}^s a_{i_k} = \sum_{k=1}^t b_{j_k}; \quad 2. \sum_{k=1}^s a_{i_k} \neq \sum_{k=1}^t b_{j_k}.$$

В первом случае при любом $\varepsilon \neq 0$ ни одно из равенств (5.22) не возможно, так как $m > t > 0$.

Положим

$$\delta' = \min \left| \sum_{k=1}^t a_{ik} - \sum_{k=1}^s b_{jk} \right|,$$

где минимум берется по всевозможным неравным между собой парам частных сумм производства и потребления задачи T . Поскольку таких пар конечное число, минимум достигается на одной из них и, следовательно, $\delta' > 0$.

Допустим, что имеет место случай 2 и

$$0 < \varepsilon < \delta = \frac{\delta'}{m}. \quad (5.23)$$

Тогда левая часть (5.22) не меньше δ' , правая же часть строго меньше δ' , т. е. равенство (5.22) невозможно и в этом случае.

Итак, при ε , удовлетворяющем неравенству (5.23), ни одна из частных сумм объемов производства задачи $T(\varepsilon)$ не совпадает ни с одной из частных сумм ее объемов потребления. Следовательно, $T(\varepsilon)$ — невырожденная транспортная задача.

Пусть $X(\varepsilon) = \|x_{ij}(\varepsilon)\|$ — произвольный опорный план задачи $T(\varepsilon)$. Разложим векторы

$$P = \begin{pmatrix} a_1 \\ \vdots \\ a_m \\ b_1 \\ \vdots \\ b_m \end{pmatrix}, \quad Q = \begin{pmatrix} 1 \\ \vdots \\ 1 \\ 0 \\ \vdots \\ m \end{pmatrix}$$

по базису плана $X(\varepsilon)$. Обозначим через $\|x_{ij}\|$ и $\|a_{ij}\|$ матрицы коэффициентов разложения вектора P и Q соответственно, дополненные нулями. Тогда

$$\|x_{ij}(\varepsilon)\| = \|x_{ij}\| + \varepsilon \|a_{ij}\|. \quad (5.24)$$

Очевидно, что

$$\begin{aligned} \sum_{j=1}^n x_{ij} &= a_i, \quad i = 1, 2, \dots, m; \\ \sum_{i=1}^m x_{ij} &= b_j, \quad j = 1, 2, \dots, n. \end{aligned} \quad (5.25)$$

В § 3 гл. 6 будет установлено существование такого положительного Δ , что при

$$0 < \varepsilon < \Delta \quad (5.26)$$

справедливы следующие утверждения:

а) если $\|x_{ij}(\varepsilon)\| = \|x_{ij} + \varepsilon a_{ij}\|$ — опорный план задачи $T(\varepsilon)$, то при условии

$$0 \leq \varepsilon' < \Delta \quad (5.27)$$

$\|x_{ij}(\varepsilon')\| = \|x_{ij} + \varepsilon' a_{ij}\|$ — опорный план задачи $T(\varepsilon')$;

б) если $\|x_{ij}(\varepsilon)\|$ — решение задачи $T(\varepsilon)$, то $\|x_{ij}(\varepsilon')\|$ — решение задачи $T(\varepsilon')$, $0 \leq \varepsilon' < \Delta$.

В частности, полагая в (5.27) $\varepsilon' = 0$, делаем вывод о том, что при условии (5.26) первая составляющая разложения (5.24) оптимального плана задачи $T(\varepsilon)$ является решением задачи T .

Допустим теперь, что нам необходимо найти решение некоторой транспортной задачи T (быть может вырожденной!). Отмеченная связь между транспортными задачами T и $T(\varepsilon)$ при достаточно малых ε [удовлетворяющих условиям (5.23), (5.26)] позволяет применить для этой цели алгоритм метода потенциалов. В самом деле, будем решать вместо задачи T вспомогательную задачу $T(\varepsilon)$, предполагая величину ε достаточно малой. Поясним последнее допущение. При решении задачи методом потенциалов необходимо сравнивать между собой величины перевозок. Пусть

$$x_{i_1 j_1}(\varepsilon) = x_{i_1 j_1} + \varepsilon a_{i_1 j_1}, \quad x_{i_2 j_2} = x_{i_2 j_2} + \varepsilon a_{i_2 j_2} —$$

любая пара перевозок задачи $T(\varepsilon)$.

Будем считать ε настолько малым, что неравенство $x_{i_1 j_1}(\varepsilon) < x_{i_2 j_2}(\varepsilon)$ выполняется при соблюдении одного из следующих условий:

либо

$$x_{i_1 j_1} < x_{i_2 j_2},$$

либо

$$x_{i_1 j_1} = x_{i_2 j_2},$$

но

$$a_{i_1 j_1} < a_{i_2 j_2}.$$

$T(\epsilon)$ при достаточно малом ϵ , как было показано, — невырожденная задача транспортировки. Поэтому метод потенциалов через конечное число итераций приводит к ее решению. Полагая теперь в полученном решении $\epsilon = 0$, получаем искомый оптимальный план задачи T .

Мы уже отмечали, что при достаточно малом ϵ_0 любой план задачи $T(\epsilon_0)$ порождает семейство планов задач $T(\epsilon)$, где $0 \leq \epsilon \leq \epsilon_0$. Все они в соответствии с (5.24) могут быть представлены единственным образом в виде

$$\|x_{ij}(\epsilon)\| = \|x_{ij}\| + \epsilon \|a_{ij}\|,$$

где $\|x_{ij}\|$, $\|a_{ij}\|$ от ϵ не зависят.

Это обстоятельство позволяет опускать при вычислении оптимального плана задачи $T(\epsilon)$ величину ϵ и оперировать лишь с составляющими плана $X(\epsilon)$

$$X = \|x_{ij}\|, \quad a = \|a_{ij}\|.$$

Введем предварительно несколько определений. Пару матриц

$$X = \|x_{ij}\|_{m,n}, \quad a = \|a_{ij}\|_{m,n}$$

назовем обобщенным планом задачи T , если при всех достаточно малых ϵ

$$X(\epsilon) = X + \epsilon a$$

является планом задачи $T(\epsilon)$. Обобщенный план удобно записывать в виде одной матрицы

$$X/a = \left\| \begin{array}{ccccccc} x_{11}/a_{11} & x_{12}/a_{12} & \dots & x_{1n}/a_{1n} & & & \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot & \cdot & \cdot & \cdot \\ x_{m1}/a_{m1} & x_{m2}/a_{m2} & \dots & x_{mn}/a_{mn} & & & \end{array} \right\|.$$

Элементы матрицы $\frac{x}{a}$ будем называть обобщенными перевозками. Между двумя любыми обобщенными перевозками введем соотношения „больше, меньше“, полагая $\frac{x_{i_1 j_1}}{a_{i_1 j_1}} > \frac{x_{i_2 j_2}}{a_{i_2 j_2}}$, если либо $x_{i_1 j_1} \triangleright x_{i_2 j_2}$,

либо $x_{l_1 j_1} = x_{l_2 j_2}$, но $\alpha_{l_1 j_1} > \alpha_{l_2 j_2}$. Очевидно, что условие

$$x_{l_1 j_1} / \alpha_{l_1 j_1} = x_{l_2 j_2} / \alpha_{l_2 j_2}$$

эквивалентно равенствам

$$x_{l_1 j_1} = x_{l_2 j_2}, \quad \alpha_{l_1 j_1} = \alpha_{l_2 j_2}.$$

Пусть X_1/α_1 , X_2/α_2 — два обобщенных плана задачи T . Будем говорить, что первый план соответствует меньшим транспортным издержкам по сравнению со вторым, если либо

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^{(1)} < \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^{(2)},$$

либо

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^{(1)} = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^{(2)},$$

но

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} \alpha_{ij}^{(1)} < \sum_{i=1}^m \sum_{j=1}^n c_{ij} \alpha_{ij}^{(2)}.$$

Оптимальным обобщенным планом T назовем такой план, который соответствует минимальным транспортным издержкам.

Элемент матрицы транспортных издержек c_{ij} будем называть $\frac{X}{\alpha}$ — существенным, если соответствующая обобщенная перевозка x_{ij}/α_{ij} положительна, т. е. $x_{ij}/\alpha_{ij} > 0$. Вычисления оптимального обобщенного плана задачи T , первая составляющая которого является ее решением, можно провести согласно алгоритму метода потенциалов, если все встречающиеся при этом понятия (такие, как план, X — существенный элемент, положительная перевозка и т. д.) рассматривать в обобщенном смысле.

Пример 2.3.

Требуется найти решение транспортной задачи

		Пункты потребления				
		1	2	3		
Пункты производства	1	2	3	4	1	Объемы производства
	2	1	5	7	1	
	3	2	4	1	1	
	.	1	1	1		
		Объемы потребления				

Определим методом северо-западного угла исходный обобщенный план

$$X_0/\alpha_0 = \left\| \begin{array}{ccc} 1/0 & 0/1 & 0 \\ 0 & 1/-1 & 0/2 \\ 0 & 0 & 1/1 \end{array} \right\|.$$

Отмечая в матрице C $\frac{X_0}{\alpha_0}$ — существенные элементы и решая соответствующие уравнения, находим

$$\begin{aligned} u_1^{(0)} &= 0, & u_2^{(0)} &= -2, & u_3^{(0)} &= 4; \\ v_1^{(0)} &= 2, & v_2^{(0)} &= 3, & v_3^{(0)} &= 5. \end{aligned}$$

Откуда

$$C_1 = \left\| \begin{array}{ccc} 0 & 0 & -1 \\ \boxed{-3} & 0 & 0 \\ 4 & 5 & 0 \end{array} \right\|.$$

Итерация I.

Этап 1 отсутствует.

Этап 2.

$$X_0/\alpha_0 = \left\| \begin{array}{ccc} 1^*/0 & 0^*/1 & 0 \\ 0 & 1^*/-1 & 0/2 \\ 0 & 0 & 1/1 \end{array} \right\| \xrightarrow[\theta_1 = \frac{1}{-1}]{1} X_1/\alpha_1 = \left\| \begin{array}{ccc} 0/1 & 1/0 & 0 \\ 1/-1 & 0 & 0/2 \\ 0 & 0 & 1/1 \end{array} \right\|.$$

Итерация II.

Этап 1.

$$C_1 = \begin{vmatrix} \bar{0} & \bar{0} & -1 \\ -\bar{3} & 0 & \bar{0} \\ 4 & 5 & \bar{0} \\ & & -3 \end{vmatrix} \xrightarrow{+3} C_2 = \begin{vmatrix} 0 & 0 & \boxed{-4} \\ 0 & 3 & 0 \\ 7 & 8 & 0 \end{vmatrix}.$$

Этап 2.

$$X_1/a_1 = \begin{vmatrix} 0^*/1 & 1/0 & 0 \\ 1^*/-1 & 0 & 0^*/2 \\ 0 & 0 & 1/1 \end{vmatrix} \xrightarrow{\theta=0/1} X_2/a_2 = \begin{vmatrix} 0 & 1/0 & 0/1 \\ 1/0 & 0 & 0/1 \\ 0 & 0 & 1/1 \end{vmatrix}.$$

Итерация III.

Этап 1.

$$C_2 = \begin{vmatrix} 0 & \bar{0} & -\bar{4} \\ \bar{0} & 3 & \bar{0} \\ 7 & 8 & \bar{0} \\ & & -4 \end{vmatrix} \xrightarrow{+4} C_3 = \begin{vmatrix} 4 & 0 & 0 \\ 0 & \boxed{-1} & 0 \\ 7 & 4 & 0 \end{vmatrix}.$$

Этап 2.

$$X_2/a_2 = \begin{vmatrix} 0 & 1/0 & 0/1 \\ 1/0 & 0 & 0/1 \\ 0 & 0 & 1/1 \end{vmatrix} \xrightarrow{\theta=0/1} X_3/a_3 = \begin{vmatrix} 0 & 1/-1 & 0/2 \\ 1/0 & 0/1 & 0 \\ 0 & 0 & 1/1 \end{vmatrix}.$$

Итерация IV.

Этап 1.

$$C_3 = \begin{vmatrix} 4 & \bar{0} & \bar{0} \\ \bar{0} & -\bar{1} & 0 \\ 7 & 4 & \bar{0} \\ & & -1 \end{vmatrix} \xrightarrow{+1} C_4 = \begin{vmatrix} 3 & 0 & 0 \\ 0 & 0 & 1 \\ 6 & 4 & 0 \end{vmatrix}.$$

Этап 2 отсутствует, т. к. X_3/a_2 — оптимальный обобщенный план.

Искомое решение транспортной задачи

$$X_3 = \begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}.$$

Заметим, что последние три итерации служили, по существу, для выяснения оптимальности плана $\begin{vmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 1 \end{vmatrix}$, который был построен в первой итерации.

В заключение дадим практическую рекомендацию для решения вырожденных задач. На практике вырожденные транспортные задачи встречаются довольно часто. Существуют важные классы транспортных задач, в которых каждый опорный план является распадающимся. Примером может послужить разбираемая в следующем параграфе проблема выбора, каждый опорный план которой содержит n положительных перевозок (вместо $2n - 1$). Однако образование цикла в транспортной задаче — явление чрезвычайной редкости. Поэтому естественно вместо довольно сложного правила, описанного в этом пункте и гарантирующего от зацикливания, использовать более простые, хотя и менее надежные, критерии вывода векторов коммуникаций из базиса. Одним из таких критериев является, например, следующий: из базиса выводится тот из «подозрительных» векторов коммуникаций P_{ij} , у которого первый индекс (i) минимальный. Приведенное правило не дает полной гарантии от зацикливания, хотя практически образование цикла почти невозможно. Если все же цикл будет получен, необходимо, начиная от дважды пройденного базиса, использовать методику, изложенную в настоящем пункте, до первого уменьшения суммарных транспортных издержек. После этого целесообразно снова вернуться к упрощенному правилу.

§ 3. ВЕНГЕРСКИЙ МЕТОД

3.1. Предварительные замечания. В предыдущем параграфе был рассмотрен метод решения задачи транспортировки, связанный с последовательными улучшениями планов перевозок (по каждому промежуточному

плану из пунктов производства все вывозилось, а пункты потребления обеспечивались полностью). Как уже отмечалось, метод потенциалов является детализацией метода последовательного улучшения плана применительно к транспортной проблеме. В этом параграфе излагается метод решения задачи транспортировки, основанный на иных соображениях. Общая схема метода такова. Вначале строится план перевозок, не удовлетворяющий, вообще говоря, всем условиям транспортной задачи T (из некоторых пунктов производства не весь продукт вывозится, часть пунктов потребления не полностью удовлетворена). Далее осуществляется переход к другому плану перевозок, более близкому к плану задачи T (новый план связан с меньшим количеством невывезенного продукта по сравнению с предыдущим). Если построенный план перевозок удовлетворяет условиям транспортной задачи (5.1), то он является оптимальным, и процесс решения на этом заканчивается. Если же некоторые условия рассматриваемой задачи остаются неудовлетворенными, метод дает возможность перейти к следующему плану перевозок, более близкому к плану задачи T . Через конечное число шагов метод приводит к плану перевозок, являющемуся планом задачи T . Как только это произойдет, процесс решения задачи заканчивается, так как построенный план перевозок автоматически совпадает с ее оптимальным планом. Метод, общая схема которого была только что приведена, обычно называют венгерским. Интересно отметить, что идея этого метода была высказана венгерским математиком Эгервари (отсюда и название метода) задолго до возникновения теории линейного программирования (1931 г.). Длительное время работа Эгервари [38] оставалась малоизвестной. Заслуга ее «открытия» принадлежит Куну, который в 1953 г. перевел ее на английский язык. В дальнейшем Кун [51] развил идею Эгервари и предложил метод, названный им венгерским, для решения проблемы выбора (частный случай транспортной задачи). В 1957 г. венгерский метод был усовершенствован Манкресом [60], который перенес его на произвольную транспортную задачу.* Если говорить о связи венгерского метода с общими методами линейного про-

* См также работы Форда и Фулкерсона [40], [41].

граммирования, то наиболее близко он примыкает к методам, опирающимся на теорию двойственности, описанию которых посвящена гл. 4. По существу этот метод, как будет показано в § 3 гл. 6, является уточнением метода последовательного сокращения невязок применительно к транспортной задаче. Венгерский метод особенно эффективен при решении им проблемы выбора. Поэтому при описании метода мы вначале остановимся на этом частном случае транспортной задачи.

3.2. Проблема выбора. Допустим, что имеется n различного рода работ $A_1, A_2, \dots, A_n$ и n человек (механизмов) $B_1, B_2, \dots, B_n$ каждый из которых может с тем или иным успехом выполнять работы $A_1, A_2, \dots, A_n$; при этом каждый работник (механизм) в состоянии выполнять лишь одну работу. Через c_{ij} обозначим величину, связанную с производительностью работника (механизма) B_i при выполнении им работы A_j . Система величин $c_{i1}, c_{i2}, \dots, c_{in}$ определяется, таким образом, квалификацией работника (качеством механизма) B_i . Естественным желанием руководителя рассматриваемых работ является такая расстановка людей (механизмов), чтобы суммарный эффект их труда (использования) был максимальным. Задачу, связанную с таким оптимальным распределением работ, принято называть проблемой выбора (или, как иногда ее называют, проблемой назначения). Математическая проблема выбора может быть сформулирована следующим образом.

Из квадратной матрицы

$$C = \begin{vmatrix} c_{11} & c_{12} & \dots & c_{1n} \\ c_{21} & c_{22} & \dots & c_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ \cdot & \cdot & \cdot & \cdot \\ c_{n1} & c_{n2} & \dots & c_{nn} \end{vmatrix}$$

выбрать такую последовательность элементов $\{c_{1j_1}, c_{2j_2}, \dots, c_{nj_n}\}$, чтобы $j_k \neq j_l$ при $k \neq l$ и при этом вели-

чина $\sum_{k=1}^n c_{kj_k}$ достигала своего максимального значения.

Другими словами, необходимо из каждой строки и каждого столбца матрицы C выбрать ровно по одному элементу (всего n элементов) так, чтобы их сумма

была наибольшей. Конечно, если бы матрица C была устроена так, что максимальные элементы ее строк лежали бы в различных столбцах, решение проблемы выбора было бы тривиальным: оптимальный выбор в этом случае имеет вид $\{c_{kj_k}\}$, где

$$c_{kj_k} = \max_{1 \leq j \leq n} c_{kj}.$$

Однако обычно несколько максимальных элементов строк расположены в одном и том же столбце и решение проблемы выбора не носит столь очевидного характера. В общем случае приходится выбирать оптимальный вариант из достаточно большого числа возможных. Действительно, как легко усмотреть, число возможных вариантов в проблеме выбора с матрицей порядка n равно $n!$, причем просмотр каждого связан с n операциями сложения и одной операцией сравнения. Поэтому при значительном n неупорядоченный перебор возможных вариантов приводит к астрономическому числу операций.

Применение венгерского метода к проблеме выбора существенно понижает количество необходимых операций. Можно показать, что общее число действий, необходимых для решения этим методом проблемы выбора порядка n , не превышает cn^3 , где c не зависит от n .

При описании метода будем придерживаться следующего порядка: вначале изложение алгоритма, затем необходимые обоснования.

3.3. Алгоритм решения проблемы выбора. Начнем с определений. Нулевые элементы квадратной матрицы D $z_1, z_2, \dots, z_k$ будем называть независимыми нулями, если для любого $1 \leq i \leq k$ строка и столбец, на пересечении которых лежит элемент z_i , не содержат элементов $z_k, k \neq i$. Две прямоугольные матрицы порядка m, n $C = \|c_{ij}\|_{m,n}$ и $D = \|d_{ij}\|_{m,n}$ назовем эквивалентными ($C \sim D$), если

$$c_{ij} = d_{ij} + \alpha_i + \beta_j, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

В процессе решения некоторые строки (столбцы) квадратной матрицы C и ей эквивалентных матриц будут выделяться значком «+», стоящим справа от соответствующей строки (над соответствующим столбцом). Элементы, лежащие в выделенных линиях (строках или

столбцах) этих матриц, будем называть выделенными элементами. Алгоритм состоит из подготовительного этапа и не более чем $(n - 2)$ последовательно проводимых итераций. Каждая итерация связана с некоторыми эквивалентными преобразованиями матрицы, полученной в результате проведения предыдущей итерации, и выбором максимального числа независимых нулей. Окончательным результатом итерации является увеличение числа независимых нулей, имеющихся в ее начале, на единицу. Как только количество независимых нулей становится равным n , проблема выбора оказывается решенной: оптимальный вариант определяется позициями независимых нулей в последней из матриц, эквивалентных C .

Подготовительный этап. Разыскиваем максимальный элемент в j -м столбце и все элементы этого столбца последовательно вычитаем из максимального. Эта операция проделывается со всеми столбцами матрицы C , $1 \leq j \leq n$. В результате образуется матрица с неотрицательными элементами, в каждом столбце которой имеется по крайней мере 1 нуль. Рассматриваем i -ю строку полученной матрицы и из каждого ее элемента вычитаем минимум этой строки. Меняя i от 1 до n , получаем матрицу C_0 с неотрицательными элементами, в каждой строке и каждом столбце которой имеется по меньшей мере один нуль. Отмечаем произвольный нуль в первом столбце звездочкой (*). Затем просматриваем второй столбец и, если в нем обнаруживается нуль, в строке которого нет нуля со звездочкой, отмечаем его звездочкой. Аналогично просматриваются один за другим все остальные столбцы матрицы C_0 . Очевидно, что нули этой матрицы, отмеченные звездочкой, являются независимыми. На этом подготовительный этап заканчивается.

Описание отдельной итерации. Допустим, что k -я итерация уже проведена и через C_k обозначена матрица, с которой мы подошли к началу $(k + 1)$ -й итерации. Если в матрице C_k звездочкой отмечено ровно n нулей, процесс решения заканчивается, причем оптимальный выбор определяется позициями нулей со звездочкой матрицы C_k . Если же число нулей со звездочкой матрицы C_k меньше n , переходим к $(k + 1)$ -й итерации, описание которой приводится ниже.

Перед началом итерации выделяем (значком $+$) столбцы C_k , содержащие нули со звездочкой.

Этап 1. Если среди невыделенных элементов матрицы C_k нет нулевых, переходим к этапу 3. Если невыделенный нуль C_k обнаруживается, возможна одна из двух альтернатив:

а) строка, содержащая этот нуль, содержит также нуль со звездочкой;

б) упомянутая строка нуля со звездочкой не содержит. В случае а) ставим над найденным нулем штрих ($'$), выделяем строку, его содержащую (ставим справа от нее значок $+$), и уничтожаем значок выделения ($+$) над столбцом, на пересечении которого с только что выделенной строкой расположен 0^* (нуль со звездочкой).

В случае б) отмечаем полученный нуль штрихом и переходим к этапу 2.

Совокупность операций, связанных с описанным исследованием одного невыделенного нуля C_k , назовем шагом этапа 1. После конечного числа шагов либо обнаружится наличие альтернативы б) (переходим к этапу 2), либо все нули окажутся выделенными (переходим к этапу 3).

Этап 2. Исходя из нуля со штрихом, в строке которого нет нуля со звездочкой [признак альтернативы б)], строим следующую цепочку элементов матрицы C_k : исходный нуль со штрихом, нуль со звездочкой (если такой найдется), лежащий с ним в одном столбце, нуль со штрихом, лежащий в одной строке с предшествующим нулем со звездочкой, и т. д. Итак, цепочка образуется движением от $0'$ к 0^* по столбцу, от 0^* к $0'$ по строке. Ниже (п. 3.4) будет показано, что описанный алгоритм построения цепочки однозначен и конечен. При этом цепочка обязательно заканчивается нулем со штрихом. Над элементами цепочки, стоящими на нечетных местах (нули со штрихом), ставим звездочки, уничтожая их над ее четными элементами. Далее уничтожаются все штрихи над элементами матрицы C_k и значки выделения ее линий. Число независимых нулей C_k увеличено на 1; $(k+1)$ -я итерация закончена, $C_{k+1} = C_k$.

Этап 3. К этому этапу переходят после завершения этапа 1 выделением всех нулей матрицы C_k . Среди невыделенных элементов C_k (ниже будет пока-

зано, что такие обязательно найдутся) выбираем минимальный: $h > 0$. Далее, величину h вычитаем из всех элементов C_k , расположенных в ее невыделенных строках, и прибавляем ко всем элементам этой матрицы, лежащим в выделенных столбцах. Получаем новую матрицу $C_k^{(1)}$ (эквивалентную C_k). Поскольку среди невыделенных элементов $C_k^{(1)}$ имеются нули, то переходим к уже описанному этапу 1, с той разницей, что C_k заменяется матрицей $C_k^{(1)}$.

Завершив этап 1, мы либо переходим к этапу 2 [имеет место альтернатива б)], либо возвращаемся к этапу 3 (все нули $C_k^{(1)}$ оказываются выделенными). В первом случае после проведения этапа 2 $(k+1)$ -я итерация заканчивается. Во втором случае в результате этапа 3 приходим к матрице $C_k^{(2)} \sim C_k^{(1)} \sim C_k$, содержащей невыделенные нули, и вся последовательность операций повторяется, отправляясь от $C_k^{(2)}$. Ниже будет показано, что после конечного числа $p - 1$ возвращений в этап 3 очередной этап 1 обязательно завершится альтернативой б). Проведя далее этап 2, мы увеличим число 0^* на 1 и, следовательно, закончим $(k+1)$ -ю итерацию. При этом

$$C_{k+1} = C_k^{(p)}.$$

3.4. Обоснование алгоритма решения проблемы выбора. Прежде всего установим связь между проблемой выбора, задаваемой матрицей C , и некоторой задачей транспортировки. Каждой последовательности элементов матрицы C вида:

$$c_{1j_1}, c_{2j_2}, \dots, c_{nj_n},$$

где $j_k \neq j_l$ при $k \neq l$, поставим в соответствие квадратную матрицу $X = \|x_{ij}\|_{nn}$, где

$$x_{ij} = \begin{cases} 0, & j \neq j_i, \\ 1, & j = j_i. \end{cases} \quad (5.28)$$

Пусть H — множество всех матриц вида (5.28) (в каждой строке и каждом столбце любой из матриц, принадлежащих H , содержится ровно одна единица, остальные элементы — нули). Проблема выбора состоит в отыскании такой матрицы $X^* \in H$, что

$$\sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^* = \max_{X \in H} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}.$$

Рассмотрим теперь транспортную задачу, состоящую в обращении в минимум

$$\sum_{i=1}^n \sum_{j=1}^n (-c_{ij}) x_{ij} \quad (5.29)$$

при условиях:

$$\sum_{i=1}^n x_{ij} = \sum_{j=1}^n x_{ij} = 1, \quad i, j = 1, 2, \dots, n; \quad (5.30)$$

$$x_{ij} \geq 0. \quad (5.31)$$

Обозначим через $\bar{H}$ множество планов задачи (5.29), (5.30), (5.31). Очевидно, что $H \in \bar{H}$ (множество матриц $\bar{H}$ включает в себя введенное ранее множество H). Следовательно,

$$\min_{x \in H} \sum_{i=1}^n \sum_{j=1}^n (-c_{ij}) x_{ij} \geq \min_{x \in \bar{H}} \sum_{i=1}^n \sum_{j=1}^n (-c_{ij}) x_{ij}.$$

Но

$$\min_{x \in H} \sum_{i=1}^n \sum_{j=1}^n (-c_{ij} x_{ij}) = -\max_{x \in H} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}.$$

Откуда

$$\max_{x \in H} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \leq -\min_{x \in \bar{H}} \sum_{i=1}^n \sum_{j=1}^n (-c_{ij}) x_{ij}. \quad (5.32)$$

Пусть X^* — опорное решение задачи (5.29), (5.30), (5.31). Из неравенства (5.32) имеем

$$\max_{x \in H} \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij} \leq \sum_{i=1}^n \sum_{j=1}^n c_{ij} x_{ij}^*. \quad (5.33)$$

В § 1 настоящей главы было показано, что любой опорный план транспортной задачи с целочисленными условиями (в нашем случае все они равны 1) состоит из целых перевозок. Поэтому из условий (5.30), (5.31), которым удовлетворяют элементы матрицы X^* , вытекает, что $X^* \in H$. Обращаясь, наконец, к неравенству (5.33), обнаруживаем, что $X^* \in H$ является решением рассматриваемой проблемы выбора. Итак, проблема выбора с матрицей C эквивалентна транспортной задаче с единичными ограничениями и матрицей транспортных издержек, равной

$$-C = \| -c_{ij} \|_{n, n}.$$

Установим одно вспомогательное утверждение, связанное с понятием эквивалентных матриц.

Рассмотрим две транспортные задачи, одна из которых определяется матрицей транспортных издержек $C = \|c_{ij}\|_{m,n}$ и вектором производства — потребления $(a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n)$, другая — матрицей $D = \|d_{ij}\|_{m,n}$ и тем же вектором. Первую задачу обозначим через T_1 , вторую — через T_2 , множество планов этих задач — через K . Предположим теперь, что матрицы C и D эквивалентны ($C \sim D$), т. е.

$$\|c_{ij}\|_{m,n} = \|d_{ij} + \alpha_i + \beta_j\|_{m,n}.$$

В таком случае решение одной из введенных задач является одновременно решением другой задачи. В самом деле, пусть план $X^* = \|x_{ij}^*\|_{m,n}$ является, например, решением задачи T_1 , т. е.

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^* = \min_{X \in K} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}. \quad (5.34)$$

Пусть далее X — произвольный план из K . В силу эквивалентности матриц C , D и условий задач T_1 , T_2 имеем

$$\begin{aligned} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} &= \sum_{i=1}^m \sum_{j=1}^n (d_{ij} + \alpha_i + \beta_j) x_{ij} = \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij} + \\ &+ \sum_{i=1}^m \alpha_i \sum_{j=1}^n x_{ij} + \sum_{j=1}^n \beta_j \sum_{i=1}^m x_{ij} = \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij} + \Delta, \end{aligned} \quad (5.35)$$

где

$$\Delta = \sum_{i=1}^m \alpha_i a_i + \sum_{j=1}^n \beta_j b_j$$

не зависит от плана X . Из (5.35) следует, что

$$\min_{X \in K} \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij} = \min_{X \in K} \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} - \Delta. \quad (5.36)$$

Учитывая соотношение (5.34), (5.35), (5.36), получаем

$$\sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij}^* = \min_{X \in K} \sum_{i=1}^m \sum_{j=1}^n d_{ij} x_{ij},$$

другими словами, X^* является оптимальным планом задачи T_2 . Займемся теперь непосредственным обоснова-

нием алгоритма решения проблемы выбора. В подготовительном этапе алгоритма строилась матрица $C_0 \sim -C$, все элементы которой были неотрицательными. Далее осуществлялся переход от матрицы C_0 к матрице C_1 , от C_1 к C_2 и так далее до тех пор, пока при некотором $l \leq n - 2$ в матрице C_l не оказывалось ровно n независимых нулей (0^*). Заметим, что согласно правилам перехода от C_k к C_{k+1} , $k = 0, 1, \dots, l - 1$ все элементы матрицы C_k неотрицательны. Пусть n независимых нулей матрицы $C_l \sim -C$ расположены в позициях $1j_1, 2j_2, \dots, nj_n$. Тогда матрица $\|x_{ij}^*\|_{n,n}$, где

$$x_{ij}^* = \begin{cases} 0, & j \neq j_l \\ 1, & j = j_l \end{cases}$$

— опорное решение транспортной задачи с матрицей транспортных издержек C_l и единичными объемами производства, потребления (любой план этой задачи приводит к неотрицательному значению соответствующей линейной формы; $\|x_{ij}^*\|_{n,n}$ обращает форму в нуль).

Учитывая эквивалентность матриц C_l и $-C$, заключаем, что $\|x_{ij}^*\|_{n,n}$ является также оптимальным планом задачи (5.29), (5.30), (5.31). Но по определению $X^* \in H$. Следовательно, план X^* соответствует оптимальному выбору для матрицы C . Итак, матрица C_l , содержащая n нулей со звездочкой, полностью определяет оптимальный выбор; выбираются те элементы C , которым соответствуют нули со звездочкой матрицы C_l .

Переходим к обоснованию отдельных этапов алгоритма.

Подготовительный этап, так же как и этап 1, дополнительных разъяснений не требует.

Этап 2. Заметим в начале, что построение цепочки из $0'$ (нулей со штрихом) и 0^* (нулей со звездочкой) проводится однозначно, так как согласно алгоритму в каждой строке C_k содержится не более одного $0'$ (после отметки нуля штрихом содержащая его строка выделяется), в каждом столбце C_k — не более одного 0^* (все эти нули независимы). Покажем, что рассматриваемая цепочка не может содержать циклов, а, следовательно, процесс построения ее конечен.

Обозначим через

$$z_{i_1 j_1}, z_{i_2 j_1}, \dots, z_{i_s j_{s-1}}, z_{i_s j_s} \quad (5.37)$$

элементы цепочки, расположенные в порядке ее построения (нули со штрихом стоят на нечетных местах, нули со звездочкой — на четных). Рассмотрим два соседних нуля со штрихом $z_{i_t j_t}$, $z_{i_{t+1} j_{t+1}}$. Очевидно, что в тот момент, когда $z_{i_t j_t}$ отмечался штрихом, j_t -й столбец C_k не был выделен. Но поскольку этот столбец содержит $0^* : z_{i_{t+1} j_t}$, первоначально он выделялся. Следовательно, значок „+“ над ним был уничтожен до отметки $z_{i_t j_t}$. Согласно этапу 1 значок выделения над столбцом j_t уничтожается одновременно с выделением i_{t+1} -й строки ($z_{i_{t+1} j_t}$ — нуль со звездочкой). Итак, отметка штрихом $z_{i_{t+1} j_{t+1}}$, совпадающая с выделением i_{t+1} -й строки, была сделана раньше отметки штрихом элемента $z_{i_t j_t}$. Если теперь рассмотреть два произвольных $0'$ из цепочки (5.37): $z_{i_\lambda j_\lambda}$, $z_{i_\mu j_\mu}$, $\lambda < \mu$, то из проведенных рассуждений с очевидностью следует, что $z_{i_\mu j_\mu}$ был отмечен (штрихом) раньше, чем $z_{i_\lambda j_\lambda}$.

Поэтому все нули со штрихом цепочки (5.37) различны. Выписанная нами цепочка обрывается нулем со штрихом. Покажем, что это не случайно: оканчиваться 0^* рассматриваемая цепочка не может. Допустим противное, т. е. существование цепочки

$$z_{i_1 j_1}, \dots, z_{i_{t+1} j_t}, \quad (*)$$

которую дальше продолжить невозможно. Поскольку $z_{i_t j_t}$ — нуль со штрихом, j_t -й столбец не выделен. С другой стороны, в этом столбце содержится нуль со звездочкой ($z_{i_{t+1} j_t}$). Следовательно, значок выделения над ним был уничтожен, что указывает на наличие в i_{t+1} -й строке нуля со штрихом. Значит, цепочка (*) может быть продолжена вопреки нашему допущению. Теперь легко установить различие всех элементов цепочки (5.37). Относительно нулей со штрихом утверждение было доказано ранее. Что касается нулей со звездочкой, то предположение о совпадении двух таких нулей приводит к выводу о совпадении двух следующих за ними нулей со штрихом (после каждого 0^* в (5.37) обязательно следует $0'$), что невозможно. Таким образом, цепочка (5.37) состоит не более чем из $2m + 1$ элементов, где m — число 0^* в матрице C_k , и оканчивается всегда

нулем со штрихом. Отметим звездочкой нечетные элементы последовательности (5.37), уничтожив предварительно стоящие над ними штрихи, и сотрем значки (*) над ее четными элементами. Очевидно, что число 0^* при этом увеличится на единицу. Покажем, что вновь полученная совокупность 0^* состоит из независимых нулей. В образованном множестве нулей со звездочкой будем различать «старые» и «новые» нули. Если z_{ij} — «старый» нуль, то он не входит в последовательность (5.37), откуда следует отсутствие в i -й строке (j -м столбце) «новых» нулей. Что касается других «старых» нулей, то i -я строка (j -й столбец) их не содержат в силу независимости предыдущего множества нулей со звездочкой. Рассмотрим произвольный «новый» нуль $z_{i_\lambda j_\lambda}$. Заметим вначале, что в его строке и столбце нет других «новых» нулей. Действительно, если бы в i_λ -й строке содержались два «новых» нуля, она обладала бы двумя нулями со штрихом, чего быть не может. С другой стороны, предположение о наличии в j_λ -м столбце двух «новых» нулей $z_{i_\lambda j_\lambda}$, $z_{i_\mu j_\mu}$ приводит к выводу о совпадении двух элементов цепочки (5.37): $z_{i_\lambda+1 j_\lambda}$, $z_{i_\mu+1 j_\mu}$, что противоречит ее структуре. Отсутствие «старых» нулей в строке i_λ и столбце j_λ уже доказано.

Итак, этап 2 приводит к увеличению числа независимых нулей на единицу.

Этап 3. Пусть все нули матрицы C_k оказались выделенными. Обозначим число нулей со звездочкой матрицы C_k через m_k (очевидно, что $m_k < n$, так как в противном случае задача была бы решена на предыдущей итерации). Заметим, что число строк и столбцов матрицы C_k , выделенных согласно алгоритму, совпадает с m_k (этап 1). Если через m_k' обозначить число выделенных строк, а через m_k'' — число выделенных столбцов, то количество невыделенных элементов матрицы C_k определяется формулой:

$$n(n - m_k) + m_k' \cdot m_k'' \geq n(n - m_k) \geq n.$$

Поэтому имеется некоторое непустое множество невыделенных элементов. Пусть I_k' (I_k'') обозначает совокупность индексов невыделенных строк (столбцов) матрицы C_k . В соответствии с алгоритмом в третьем

этапе осуществляется переход от матрицы C_k к матрице C_k' , определяемой следующими соотношениями:

$$c_{ij}^{(k,1)} = \begin{cases} c_{ij}^{(k)} - h, & i \in I_k', \quad j \in I_k'', \\ c_{ij}^{(k)} + h, & i \in \bar{I}_k', \quad j \in \bar{I}_k'', \\ c_{ij}^{(k)}, & i \in I_k', \quad j \in \bar{I}_k'' \text{ или} \\ & i \in \bar{I}_k', \quad j \in I_k''. \end{cases} \quad (5.38)$$

Здесь через h обозначается $\min_{\substack{i \in I_k' \\ j \in I_k''}} c_{ij}^{(k)}$.

Вспомним, что согласно алгоритму (этап 1) все отмеченные нули (звездочкой или штрихом) матрицы C_k расположены либо в выделенной строке, либо в выделенном столбце (ни один из них не лежит на пересечении выделенных строки и столбца). Поэтому согласно (5.38) все отмеченные нули матрицы C_k переходят в нули матрицы C_k^1 (звездочки и штрихи при этом не стираются). Все элементы C_k^1 остаются неотрицательными, поскольку на h уменьшаются лишь невыделенные элементы C_k , а h по определению является минимальным из них. Кроме того, среди невыделенных элементов C_k^1 по меньшей мере один совпадает с нулем. Продолжая процесс вычисления, мы либо переходим к этапу 2 и увеличиваем число независимых нулей на единицу, либо, выделив все нулевые элементы C_k^1 , возвращаемся снова к этапу 3. При этом, хотя общее число выделенных линий C_k^1 остается равным m_k , число выделенных строк увеличивается по крайней мере на единицу (вновь образованные нули C_k^1 согласно алгоритму должны быть расположены в выделенных строках, старые же выделенные строки остаются без изменения). Проведя еще раз этап 3, переходим к матрице $C_k^{(2)} \sim C_k^{(1)} \sim C_k$ и либо увеличиваем число независимых нулей на 1, либо выделяем все нули $C_k^{(2)}$, увеличив при этом число выделенных строк за счет уменьшения числа выделенных столбцов. Последовательный переход от C_k к $C_k^{(1)}$, от $C_k^{(1)}$ к $C_k^{(2)}$, и так далее осуществляется до тех пор, пока для неко-

торой матрицы $C_k^{(p)}$ после этапа 1 не последует этап 2. Покажем, что

$$p \leq m_k + 1. \tag{5.39}$$

Действительно при переходе от $C_k^{(\mu)}$ к $C_k^{(\mu+1)}$, $\mu = 0, 1 \dots, p - 1$ число выделенных строк увеличивается не менее чем на 1. Поэтому, если после m_k циклов (этап 3 + этап 1) не последовало этапа 2, все нули $C_k^{(m_k)}$ расположены в m_k выделенных строках этой матрицы. Поскольку в любой строке, содержащей «новый» нуль матрицы $C_k^{(m_k+1)}$, нет нулей со звездочкой (все 0^* расположены в выделенных строках), мы сразу же переходим к этапу 2. Неравенство (5.39) доказано. Этим завершается обоснование алгоритма. Для иллюстрации алгоритма решения проблемы выбора приведем численный пример.

Пример. Решить проблему выбора, определяемую матрицей

$$C = \begin{vmatrix} 3 & 4 & 2 & 2 & 1 \\ 4 & 5 & 3 & 1 & 3 \\ 4 & 3 & 1 & 1 & 1 \\ 3 & 1 & 2 & 2 & 2 \\ 1 & 3 & 1 & 2 & 1 \end{vmatrix}.$$

При использовании алгоритма переход от одной матрицы к другой (после 2-го или 3-го этапов) отмечается стрелкой, под которой помещается соответствующая величина h (в случае этапа 3). Значок выделения, подлежащий уничтожению, обводится кружком. Цепочка в этапе 2 намечается стрелками.

$C = \begin{vmatrix} 3 & 4 & 2 & 2 & 1 \\ 4 & 5 & 3 & 1 & 3 \\ 4 & 3 & 1 & 1 & 1 \\ 3 & 1 & 2 & 2 & 2 \\ 1 & 3 & 1 & 2 & 1 \end{vmatrix}$

$\longrightarrow$
 подготовитель-
 ный этап

$C_0 = \begin{vmatrix} 1 & 1 & 1 & 0^* & 2 \\ 0^* & 0^* & 0 & 1 & 0 \\ 0^* & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \end{vmatrix}$

$\longrightarrow$

$\oplus$
 $+$

этапы 1, 2
 Итерация 1

$$\rightarrow C_0 = \begin{array}{c} + \quad \oplus \quad + \\ \left\| \begin{array}{ccccc} 1 & 1 & 1 & 0^* & 2 \\ 0 & 0^* & 0' & 1 & 0 \\ 0^* & 2 & 2 & 1 & 2 \\ 1 & 4 & 1 & 0 & 1 \\ 3 & 2 & 2 & 0 & 2 \end{array} \right\| \end{array} \xrightarrow[\text{этап 3}]{h=1} C_1 = \begin{array}{c} + \quad \oplus \\ \left\| \begin{array}{ccccc} 1 & 0' & 0 & 0^* & 1 \\ 1 & 0^* & 0' & 2 & 0 \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0' & 0 & 0 \\ 3 & 1 & 1 & 0 & 1 \end{array} \right\| \end{array} \begin{array}{c} + \\ + \\ \rightarrow \end{array}$$

| этап 1 | | этапы 1, 2 |

Итерация II

$$\rightarrow C_1 = \begin{array}{c} + \quad \oplus \quad \oplus \quad \oplus \\ \left\| \begin{array}{ccccc} 1 & 0' & 0 & 0^* & 1 \\ 1 & 0^* & 0 & 2 & 0' \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0^* & 0' & 0 \\ 3 & 1 & 1 & 0' & 1 \end{array} \right\| \end{array} \xrightarrow{\quad} C_1 = \begin{array}{c} \left\| \begin{array}{ccccc} 1 & 0^* & 0 & 0 & 1 \\ 1 & 0 & 0 & 2 & 0^* \\ 0^* & 1 & 1 & 1 & 1 \\ 1 & 3 & 0^* & 0 & 0 \\ 3 & 1 & 1 & 0^* & 1 \end{array} \right\| \end{array}$$

| этапы 1, 2 |

Итерация III

Отсюда оптимальный выбор имеет вид

$$\{a_{12}, a_{25}, a_{31}, a_{43}, a_{54}\}.$$

Соответствующая ему сумма

$$a_{12} + a_{25} + a_{31} + a_{43} + a_{54} = 15.$$

Сделаем пояснения к решению примера.

Процесс отыскания максимального выбора складывается из подготовительного этапа и трех итераций.

Подготовительный этап. Максимальный элемент первого столбца матрицы C равен 4. Поэтому для получения первого столбца матрицы C' необходимо из 4 вычесть элементы столбца 1 матрицы C . Аналогично, для образования 2, 3, 4 и 5 столбцов матрицы C' необходимо элементы соответствующих столбцов матрицы C вычесть из величин 5, 3, 2, 3. Поскольку в каждой строке матрицы C' имеются нулевые элементы, $C' = C'' = C_0$. В первом столбце C_0 отмечается звездочкой нуль, расположенный во второй строке. Во втором столбце матрицы C_0 единственный нуль лежит во второй строке и, следовательно, звездочкой не отмечается. Подобная ситуация имеет место и для столбцов 3, 5. В четвертом столбце отмечаем звездочкой нуль, расположенный в первой строке (в первой строке матрицы C_0 нет нулей со звездочкой). В результате предварительного этапа получено два независимых нуля. Следовательно, для решения задачи необходимо проведение трех итераций.

Итерация I. Этап 1. Выделяем (значком +) столбцы матрицы C_0 , содержащие нули со звездочкой (первый и четвертый). Переходим к просмотру невыделенных нулей C_0 , начиная со второго столбца. Отмечаем штрихом нуль этого столбца, лежащий во второй строке. Поскольку в этой строке имеется 0^* [альтернатива а)], строка подлежит выделению. Одновременно значок выделения над первым столбцом, содержащим 0^* во второй строке, уничтожается (обводится кружком). Обращаемся к невыделенным нулям первого столбца. Отмечаем штрихом нуль этого столбца, лежащий в третьей строке. Третья строка матрицы C_0 не содержит нуля со звездочкой. Следовательно, имеет место альтернатива б), указывающая на необходимость перехода к этапу 2. Этап 1, таким образом, состоит из двух шагов.

Этап 2. От последнего нуля со штрихом (третья строка, первый столбец) движемся по столбцу до нуля со звездочкой (первый столбец, вторая строка). Затем от нуля со звездочкой, лежащего на пересечении второй строки и первого столбца переходим к нулю со штрихом, расположенному в той же строке второго столбца. Поскольку во втором столбце матрицы C_0 нет 0^* , процесс образования цепочки закончен. Искомая цепочка состоит из трех элементов: $0_{31}'$, 0_{21}^* , $0_{22}'$ (индексы при нулях указывают на их положение в матрице C). Для завершения этапа 2, а вместе с ним и итерации 1 остается поставить звездочки над нулями цепочки, отмеченными штрихами, стереть звездочку над единственным четным элементом цепочки и уничтожить все значки выделения. В результате итерации 1 число независимых нулей, увеличившись на 1, стало равным 3.

Итерация II. Этап 1. Выделяем столбцы, содержащие нули со звездочкой (они имеют номера 1, 2, 4). Просмотр нуля, расположенного во второй строке первого из невыделенных столбцов, приводит к выделению этой строки и уничтожению значка + над столбцом 2 [альтернатива а)]. После этого все нулевые элементы матрицы C_0 оказываются выделенными. Итак, в данном случае этап 1 завершается переходом к этапу 3.

Этап 3. Минимальным из числа невыделенных элементов матрицы C_0 является 1. Поэтому из элементов 1, 3, 4 и 5 строк матрицы C_0 вычитается $h=1$, к элементам столбцов C_0 с номерами 1, 4 $h=1$ прибавляется.

В результате получаем матрицу $C_1 \sim C_0$. Этап 3 закончен. Переносим все значки (+, *, ') с матрицы C_0 на матрицу C_1 и переходим к этапу 1 (естественно, что значки выделения, обведенные кружком, переносить не имеет смысла).

Этап 1. В первых двух столбцах C_1 нет невыделенных нулей. Поэтому обращаемся к третьему столбцу и отмечаем штрихом его единственный невыделенный нуль. Поскольку в строке, содержащей этот нуль, нет нулей со звездочкой, необходим переход к этапу 2 [альтернатива б)].

Этап 2, завершающий итерацию II, чрезвычайно прост, так как соответствующая цепочка состоит всего лишь из одного элемента $0'_{43}$. Итерация III, в результате которой мы приходим к искомому решению, составляется из этапов 1, 2. Подробного описания этой итерации мы приводить не будем.

3.5. Общая транспортная задача, алгоритм. Описанный в п. 3.4 метод решения проблемы выбора без значительных изменений может быть перенесен на случай общей транспортной задачи. Итак, пусть требуется решить транспортную задачу (5.1), (5.2), (5.3), объемы производства и потребления которой удовлетворяют условию баланса

$$\sum_{i=1}^m a_i = \sum_{j=1}^n b_j.$$

Такую задачу мы обозначали буквой T . При описании алгоритма решения задачи T величины $a_i, b_j, i = 1, 2, \dots, m, j = 1, 2, \dots, n$ будем предполагать целыми числами*.

Алгоритм состоит из предварительного этапа и конечного числа итераций.

В результате предварительного этапа строится матрица $X_0 = \|x_{ij}^{(0)}\|_{m,n}$, элементы которой неотрицательны и удовлетворяют неравенствам:

$$\begin{aligned} \sum_{j=1}^n x_{ij}^{(0)} &\leq a_i, \quad i = 1, 2, \dots, m, \\ \sum_{i=1}^m x_{ij}^{(0)} &\leq b_j, \quad j = 1, 2, \dots, n. \end{aligned} \tag{5.40}$$

* Все рассуждения останутся в силе и при рациональных величинах объемов производства и потребления.

Если все условия (5.40) — равенства, то матрица X_0 автоматически является решением задачи T . Если же среди условий (5.40) имеются строгие неравенства, т. е. X_0 не является планом задачи T , переходим к 1 итерации.

Через $X_k = \|x_{ij}^{(k)}\|_{m,n}$ обозначим матрицу, построением которой заканчивается k -я итерация (подготовительному этапу приписываем индекс 0).

Положим

$$\Delta_k = \sum_{i=1}^m a_i + \sum_{j=1}^n b_j - 2 \sum_{i=1}^m \sum_{j=1}^n x_{ij}^{(k)}. \quad (5.41)$$

Величина Δ_k характеризует близость матрицы X_k к плану исследуемой задачи T . Проведя 1-ю итерацию, мы построим матрицу $X_1 = \|x_{ij}^{(1)}\|_{m,n}$, состоящую из неотрицательных элементов, подчиненных системе неравенств (5.40). При этом матрица X_1 такова, что $\Delta_1 < \Delta_0$. Если Δ_1 оказывается равным нулю, X_1 — оптимальный план задачи T . Если же $\Delta_1 > 0$, необходимо приступить к следующей итерации. Итерации проводятся до тех пор, пока при некотором d Δ_d окажется равным нулю. В этом случае матрица X_d является решением транспортной задачи. Поскольку величины Δ_k монотонно убывают с ростом k и являются целыми числами, $\Delta_k - \Delta_{k+1} \geq 1$, $k = 0, 1, \dots$. Следовательно, найдется такое d , что $\Delta_d = 0$. Таким образом, решение задачи T отыскивается в результате конечного числа итераций.

Перейдем теперь к детальному рассмотрению отдельных элементов алгоритма.

Подготовительный этап. В каждом из столбцов матрицы транспортных издержек $C = \|c_{ij}\|_{m,n}$ отыскиваем минимальный элемент, который затем вычитаем из всех элементов этого столбца. Получаем матрицу C' . Далее, в каждой строке C' выбираем минимальный элемент и вычитаем его из всех элементов рассматриваемой строки. Приходим к матрице $C'' = C_0 = \|c_{ij}^{(0)}\|_{m,n}$, все элементы которой неотрицательны, причем в каждой строке (каждом столбце) этой матрицы имеется по крайней мере один нуль. Затем определяем матрицу $X_0 = \|x_{ij}^{(0)}\|_{m,n}$, построение которой осуществляется по столбцам. Пусть i_k, j , $k = 1, \dots, r_j$ — номер

строки, в которой расположен k -й нуль j -го столбца матрицы C_0 (отсчет нулей производится сверху вниз). Тогда элементы первого столбца X_0 определяются рекуррентной формулой

$$x_{i1}^{(0)} = \begin{cases} 0, & i \neq i_{\lambda, 1}, \quad \lambda = 1, 2, \dots, r_1, \\ \min \left\{ a_i, b_1 - \sum_{\mu=1}^{i-1} x_{\mu 1}^{(0)} \right\}, & i = i_{\lambda, 1}, \quad \lambda = 1, \dots, r_1. \end{cases}$$

Допустим, что столбцы матрицы X_0 до $(j-1)$ -го включительно уже заполнены. Элементы j -го столбца определяются в соответствии с формулой

$$x_{ij}^{(0)} = \begin{cases} 0, & i \neq i_{\lambda, j}, \quad \lambda = 1, 2, \dots, r_j, \\ \min \left\{ a_i - \sum_{\mu=1}^{j-1} x_{i\mu}^{(0)}, b_j - \sum_{\mu=1}^{i-1} x_{\mu j}^{(0)} \right\}, & i = i_{\lambda, j}, \quad \lambda = 1, \dots, r_j. \end{cases}$$

Если

$$\Delta_0 = \sum_{i=1}^m a_i + \sum_{j=1}^n b_j - 2 \sum_{i=1}^m \sum_{j=1}^n x_{ij}^{(0)} = 0,$$

X_0 — решение задачи T (см. обоснование). Если же $\Delta_0 > 0$, переходим к первой итерации.

Описание отдельной итерации. Каждая итерация алгоритма состоит, вообще говоря, из трех качественно различных этапов. Начинается итерация этапом 1. Затем в общем случае несколько раз проводится пара этапов 3, 1 (может случиться, что итерация не содержит ни одной такой пары). Заканчивается итерация этапом 2.

Допустим, что уже проведено k итераций, причем $\Delta_k > 0$. В таком случае необходимо, отправляясь от матриц X_k, C_k , полученных в результате k -й итерации, провести следующую, $(k+1)$ -ю итерацию. Перед началом итерации выделяются (знаком $+$) те столбцы матрицы C_k , для которых

$$b_j - \sum_{i=1}^m x_{ij}^{(k)} = 0.$$

Этап 1. Если все нулевые элементы C_k оказываются выделенными, переходим к этапу 3. В противном случае выбираем произвольный невыделенный нуль ма-

трицы C_k , расположенный, скажем, в i -й строке и j -м столбце, и составляем величину

$$\delta_i^{(k)} = a_i - \sum_{\mu=1}^n x_{i\mu}^{(k)}.$$

Возможна одна из следующих альтернатив:

$$\text{а) } \delta_i^{(k)} = 0; \quad \text{б) } \delta_i^{(k)} > 0.$$

В случае а) i -я строка матрицы C_k выделяется (знаком „+“, стоящим справа от нее), элемент $c_{ij}^{(k)}$ отмечается штрихом. Далее просматриваются все выделенные столбцы C_k . Если μ -й столбец C_k выделен и $x_{i\mu}^{(k)} > 0$, значок выделения над этим столбцом уничтожается, а элемент $c_{i\mu}^{(k)}$ отмечается звездочкой. В случае б) элемент $c_{ij}^{(k)}$ отмечается штрихом, после чего осуществляется переход к этапу 2. Назовем совокупность операций, связанных с просмотром одного невыделенного нуля матрицы C_k , шагом этапа 1. Допустим, что несколько подобных шагов уже проделано, причем последний шаг завершился альтернативой а). Если все нули матрицы C_k оказываются выделенными, переходим к этапу 3. Если же среди нулей C_k имеются невыделенные, продолжаем этап 1, переходя к следующему его шагу. Итак, этап 1 складывается из нескольких шагов и завершается либо обнаружением на некотором из них альтернативы б) (переход к этапу 2), либо выделением всех нулей матрицы C_k (переход к этапу 3).

В зависимости от исхода этапа 1 будем различать случаи 1А (переход к этапу 2) и 1В (переход к этапу 3).

Этап 2 следует за первым этапом в случае 1А и состоит в построении цепочки из нулей матрицы C_k , отмеченных штрихами или звездочками, с последующим переходом к новой матрице X_{k+1} . Пусть для некоторого невыделенного нуля матрицы C_k , расположенного, скажем, в позиции λ_1, μ_1 , имеет место альтернатива б). Исходя из этого элемента, строим цепочку из отмеченных нулей матрицы C_k , руководствуясь следующим правилом. В μ_1 -м столбце C_k выбираем нуль со звездочкой — $0_{\lambda_2\mu_1}^*$ (индексы при нуле указывают на его положение в матрице C_k). В λ_2 -й строке выбираем нуль со штри-

хом — $0'_{\lambda_2 \mu_2}$. Далее от $0'_{\lambda_2 \mu_2}$ движемся по μ_2 -му столбцу до нуля со звездочкой и т. д. Последовательный переход от нуля со штрихом к нулю со звездочкой по столбцу, от нуля со звездочкой к нулю со штрихом по строке осуществляется до тех пор, пока это возможно. Можно показать (см. обоснование алгоритма), что построение цепочки согласно указанному правилу проводится однозначно, цепочка не имеет замкнутых циклов (все ее элементы различны) и всегда оканчивается нулем со штрихом.

После того, как цепочка

$$0'_{\lambda_1 \mu_1}, 0^*_{\lambda_2 \mu_1}, \dots, 0^*_{\lambda_s \mu_{s-1}}, 0'_{\lambda_s \mu_s}, \quad (5.42)$$

составленная из отмеченных нулей матрицы C_k , построена, переход от матрицы X_k к X_{k+1} осуществляется согласно следующим формулам:

$$x_{ij}^{(k+1)} = \begin{cases} x_{ij}^{(k)}, & \text{если } c_{ij}^{(k)} \text{ не входит в цепочку (5.42),} \\ x_{ij}^{(k)} + \theta_k, & \text{если } c_{ij}^{(k)} \text{ — нечетный элемент цепочки (5.42),} \\ x_{ij}^{(k)} - \theta_k, & \text{если } c_{ij}^{(k)} \text{ — четный элемент цепочки (5.42).} \end{cases} \quad (5.43)$$

Здесь через θ_k обозначен

$$\min_{2 \leq t \leq s} \left\{ x_{\lambda_t \mu_{t-1}}^{(k)}, \delta_{\lambda_1}^{(k)}, \bar{\delta}_{\mu_s}^{(k)} \right\},$$

где

$$\delta_i^{(k)} = a_i - \sum_{j=1}^n x_{ij}^{(k)}, \quad \bar{\delta}_j^{(k)} = b_j - \sum_{i=1}^m x_{ij}^{(k)}, \\ i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n.$$

Нетрудно заметить (см. обоснование), что элементы матрицы X_{k+1} неотрицательны и удовлетворяют неравенствам (5.40). При этом $\Delta_{k+1} < \Delta_k$.

Этап 3. Допустим, что этап 1 завершился случаем 1В (все нули C_k выделены). Тогда необходимо проведение этапа 3, состоящего в переходе от матрицы $C_k = \|c_{ij}^{(k)}\|$ к матрице $C_k^{(1)} = \|c_{ij}^{(k,1)}\| \sim C_k$, в которой имеется по крайней мере один невыделенный нуль, причем $c_{ij}^{(k,1)} \geq 0$ (все старые значки выделения, естественно,

сохраняются). Пусть $h = \min c_{ij}^{(k)}$, где минимум берется по всем невыделенным элементам матрицы C_k . Из элементов матрицы C_k , принадлежащих невыделенным строкам, вычитаем h , к элементам, принадлежащим выделенным столбцам, h прибавляем. Полученную матрицу обозначаем через $C_k^{(1)}$. Далее переходим к этапу 1, заменив C_k матрицей $C_k^{(1)}$.

После проведения этапа 1 мы либо перейдем к этапу 2 (1А), либо вынуждены будем снова вернуться к этапу 3 (1В).

Циклы, каждый из которых составлен из этапов 3, 1, необходимо проводить до тех пор, пока один из этапов 1 не завершится переходом к этапу 2 (случай 1А). В дальнейшем (см. обоснование) будет показано, что через конечное число p циклов это обязательно случится и, следовательно, после проведения этапа 2 $(k+1)$ -я итерация закончится. Если Δ_{k+1} оказывается равным нулю, X_{k+1} — искомое решение задачи T . Если же $\Delta_{k+1} > 0$, переходим к $(k+2)$ -й итерации, отправляясь от матриц $C_{k+1} = C_k^{(p)}$, X_{k+1} .

3.6. Обоснование алгоритма решения общей транспортной задачи. Обоснование алгоритма ничем существенным не отличается от проведенного выше обоснования метода решения проблемы выбора. Поэтому мы проведем его кратко, выделяя лишь те моменты, которые присущи задаче транспортировки в общем виде. Предварительный этап, так же как и этап 1, дополнительных обоснований не требует.

Этап 2. Обоснование построения цепочки

$$0'_{\lambda_1 \mu_1}, \quad 0^*_{\lambda_2 \mu_1}, \dots, 0'_{\lambda_s \mu_s} \quad (5.42)$$

проводится аналогично тому, как это делалось в проблеме выбора.

Действительно, однозначность формирования цепочки (5.42) следует из того, что в каждом столбце (каждой строке) C_k не более одного нуля со звездочкой (со штрихом). Отсутствие циклов в (5.42) (конечность цепочки) вытекает из разновременности отметок (звездочкой или штрихом) нулей, составляющих цепочку: при любых $t_1, t_2, t_2 \geq 1$, t_1 -й нуль цепочки отмечался позже $(t_1 + t_2)$ -го ее нуля. Далее, цепочка (5.42) обязательно

обрывается нулем со штрихом, так как в строке любого нуля со звездочкой имеется нуль со штрихом. Перейдем к выяснению свойств матрицы X_{k+1} , составленной по формулам (5.43).

Согласно формуле (5.43) при построении X_{k+1} уменьшаются лишь те элементы матрицы X_k , которые соответствуют четным элементам цепочки (5.42) (нулям со звездочкой), причем на величину, не превосходящую минимума уменьшаемых элементов. Значит

$$x_{ij}^{(k+1)} \geq 0.$$

Далее,

$$\left. \begin{aligned} \sum_{j=1}^n x_{ij}^{(k)} &= \sum_{j=1}^n x_{ij}^{(k+1)}, & i \neq \lambda_1, \\ \sum_{i=1}^m x_{ij}^{(k)} &= \sum_{i=1}^m x_{ij}^{(k+1)}, & j \neq \mu_s. \end{aligned} \right\} \quad (5.44)$$

В самом деле, в каждой строке матрицы C_k , отличной от λ_1 , так же как и в каждом ее столбце, отличном от μ_s , либо вовсе нет элементов цепочки (5.42), либо имеются два таких элемента, которые в цепочке (5.42) стоят рядом. Поэтому при образовании матрицы X_{k+1} в i -й строке, $i \neq \lambda_1$ (в j -м столбце, $j \neq \mu_s$), матрицы X_k либо все элементы остаются неизменными, либо один уменьшается на θ_k , а другой при этом увеличивается на ту же величину θ_k , т. е. соотношения (5.44) действительно имеют место. Заметим также, что

$$\left. \begin{aligned} \sum_{j=1}^n x_{\lambda_1 j}^{(k+1)} &= \sum_{j=1}^n x_{\lambda_1 j}^{(k)} + \theta_k, \\ \sum_{i=1}^m x_{i \mu_s}^{(k+1)} &= \sum_{i=1}^m x_{i \mu_s}^{(k)} + \theta_k. \end{aligned} \right\} \quad (5.45)$$

Из неравенства

$$\theta_k \leq \min \left\{ \delta_{\lambda_1}^{(k)}, \bar{\delta}_{\mu_s}^{(k)} \right\},$$

где

$$\delta_{\lambda_1}^{(k)} = a_{\lambda_1} - \sum_{j=1}^n x_{\lambda_1 j}^{(k)}, \quad \bar{\delta}_{\mu_s}^{(k)} = b_{\mu_s} - \sum_{i=1}^m x_{i \mu_s}^{(k)},$$

и соотношений (5.44), (5.45) следует, что элементы матрицы X_{k+1} удовлетворяют системе неравенств (5.40)

Используя равенства (5.44), (5.45), имеем

$$\Delta_{k+1} = \sum_{i=1}^m a_i + \sum_{j=1}^n b_j - 2 \sum_{i=1}^m \sum_{j=1}^n x_{ij}^{(k+1)} = \sum_{i=1}^m a_i + \\ + \sum_{j=1}^n b_j - 2 \sum_{i=1}^m \sum_{j=1}^n x_{ij}^{(k)} - 2\theta_k = \Delta_k - 2\theta_k.$$

Но $\theta_k > 0$, поскольку при $3 \leq t \leq s$ $x_{\lambda_t \mu_{t-1}} > 0$ ($c_{\lambda_t \mu_{t-1}}^{(k)}$ — нуль со звездочкой), $\delta_{\lambda_1} > 0$ (условие перехода к этапу 2), $\bar{\delta}_{\mu_s} > 0$ [условие обрыва цепочки (5.45)].
Значит

$$0 \leq \Delta_{k+1} < \Delta_k.$$

Этап 3. После проведения этапа 3 (перехода к матрице $C_k^{(1)}$) образуется по крайней мере один невыделенный нуль. Следующий далее этап 1 либо приведет к этапу 2, либо увеличит число выделенных строк в матрице $C_k^{(1)}$. Но так как число строк в каждой из матриц $C_k^{(t)}$ равно m , через конечное число пар этапов 1, 3 очередной этап 1 завершится случаем 1А и после проведения этапа 2 $(k+1)$ -я итерация будет окончена. Поскольку каждая итерация уменьшает сумму невязок Δ_k по меньшей мере на 2 (здесь мы используем предположение о целочисленности объемов производства и потребления задачи T)^{*)}, через конечное число d таких итераций величина Δ_d окажется равной нулю. Покажем, что X_d — оптимальный план задачи T .

Поскольку при любом k элементы матрицы X_k удовлетворяют системе неравенств (5.40),

$$\delta_i^{(k)} \geq 0, \quad \bar{\delta}_j^{(k)} \geq 0.$$

Следовательно, из равенств

$$\Delta_d = 2 \sum_{i=1}^m \delta_i^{(d)} = 2 \sum_{j=1}^n \bar{\delta}_j^{(d)} = 0$$

получаем

$$\delta_i^{(d)} = a_i - \sum_{j=1}^n x_{ij}^{(d)} = 0, \quad i = 1, 2, \dots, m, \\ \bar{\delta}_j^{(d)} = b_j - \sum_{i=1}^m x_{ij}^{(d)} = 0, \quad j = 1, 2, \dots, n.$$

^{*)} θ_k — целое положительное число. Следовательно, $\Delta_{k+1} = \Delta_k - 2\theta_k \leq \Delta_k - 2$.

Учитывая неотрицательность элементов матрицы X_d (она была доказана при обосновании этапа 2), делаем вывод о том, что X_d — план исследуемой задачи T . Прежде чем приступить к доказательству оптимальности плана X_d , сделаем два вспомогательных замечания.

Замечание 1. При любом r из положительности $x_{ij}^{(r)}$ следует, что $c_{ij}^{(r)} = 0$. Для $r = 0$ справедливость утверждения следует из способа построения матрицы X_0 (см. предварительный этап). Покажем, что при переходе от k к $k + 1$ его справедливость не может быть нарушена. Допустим, что в $(k + 1)$ -й итерации этап 3 используется p раз. Если $x_{ij}^{(k)} > 0$, то по предположению индукции элемент $c_{ij}^{(k)} = 0$ и расположен либо в выделенной строке, либо в выделенном столбце C_k (см. этап 1). Поэтому $c_{ij}^{(k, 1)} = c_{ij}^{(k)} = 0$ (см. этап 3). Аналогичные рассуждения приводят к системе равенств

$$c_{ij}^{(k)} = c_{ij}^{(k, 1)} = \dots = c_{ij}^{(k, p)} = c_{ij}^{(k+1)} = 0.$$

Заметим также, что элементы $C_k^{(p)} = C_{k+1}$, входящие в цепочку, составленную для построения матрицы X_{k+1} , равны нулю. Пусть теперь $x_{ij}^{(k+1)} > 0$. Рассмотрим два случая:

а) $x_{ij}^{(k)} > 0$; б) $x_{ij}^{(k)} = 0$.

В случае а), как было показано, $c_{ij}^{(k+1)} = 0$.

Если же имеет место б), $c_{ij}^{(k+1)}$ является, очевидно, элементом цепочки (нечетным) и, следовательно, также совпадает с нулем.

Замечание 2. Любая из матриц C_k состоит из неотрицательных элементов.

Это утверждение следует из того, что согласно алгоритму (см. этап 3) при переходе от одной матрицы к другой из уменьшаемых элементов первой матрицы вычитается их минимум.

Перейдем к установлению оптимальности плана X_d .

Рассмотрим вспомогательную транспортную задачу с матрицей транспортных издержек C_d и прежними

объемами производства и потребления. Поскольку $c_{ij}^{(d)} \geq 0$ (замечание 2), для любого плана X

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij}^{(d)} x_{ij} \geq 0. \quad (5.46)$$

С другой стороны, в силу замечания 1

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij}^{(d)} x_{ij}^{(d)} = 0. \quad (5.47)$$

Сравнивая соотношения (5.46), (5.47), делаем вывод о том, что X_d — решение вспомогательной задачи. Но согласно правилам перехода от C_k к C_{k+1} (этап 3) матрица $C \sim C_d$. Поэтому план X_d является оптимальным и для исследуемой задачи T . Обоснование венгерского метода закончено полностью.

3.7. Пример. Требуется найти решение транспортной задачи с матрицей транспортных издержек

$$C = \begin{vmatrix} 2 & 5 & 9 & 5 \\ 8 & 3 & 5 & 8 \\ 7 & 3 & 1 & 4 \\ 5 & 9 & 7 & 2 \end{vmatrix}$$

и вектором производства—потребления.

$$(a_1, a_2, a_3, a_4; b_1, b_2, b_3, b_4) = (4, 2, 3, 3; 3, 6, 2, 1).$$

Предварительный этап.

а)

$$C = \begin{vmatrix} 2 & 5 & 9 & 5 \\ 8 & 3 & 5 & 8 \\ 7 & 3 & 1 & 4 \\ 5 & 9 & 7 & 2 \end{vmatrix} \rightarrow C_0 = \begin{vmatrix} 0 & 2 & 8 & 3 \\ 6 & 0 & 4 & 6 \\ 5 & 0 & 0 & 2 \\ 3 & 6 & 6 & 0 \end{vmatrix}.$$

Матрица C_0 получена из C вычитанием чисел 2, 3, 1, 2 из элементов столбцов с номерами 1, 2, 3, 4 соответственно.

б)

$$X_0 = \begin{array}{c} \left\| \begin{array}{cccc} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right\| \begin{array}{c} 4 \\ 2 \\ 3 \\ 3 \end{array} \\ \hline \begin{array}{cccc} 3 & 6 & 2 & 1 \\ 0 & 1 & 2 & 0 \end{array} \end{array}$$

Во 2-м столбце справа от X_0 записаны величины невязок для пунктов производства.

Во 2-й строке ниже X_0 выписаны величины невязок для пунктов потребления $\frac{1}{2} \Delta_0 = 3$.

Итерация I.

$$C_0 = \begin{array}{c} \times \left\| \begin{array}{cccc} 0 & 2 & 8 & 3 \\ 6 & 0' & 4 & 6 \\ 5 & 0' & 0 & 2 \\ 3 & 6 & 6 & 0 \end{array} \right\| \begin{array}{c} \pm \\ + \\ + \\ \times \end{array} \end{array} \xrightarrow{h=2} C_0^{(1)} = \begin{array}{c} \times \left\| \begin{array}{cccc} 0 & 0' & 6 & 3 \\ 8 & 0' & 4 & 8 \\ 7 & 0' & 0 & 4 \\ 3 & 4 & 4 & 0 \end{array} \right\| \begin{array}{c} \pm \\ + \\ + \\ \times \end{array} \end{array} = C_1$$

Для удобства проведения этапа 1 строки матриц C_0 , $C_0^{(1)}$, которым соответствуют согласно X_0 положительные невязки, отмечаются значками $\times$, стоящими слева от них. Элементы матриц C_0 , C_1 , лежащие в выделенных столбцах и соответствующие положительным элементам X_0 , отмечаются черточкой. Предварительно выделяем столбцы 1, 4.

Этап 1 (1) состоит из двух шагов: отмечены строки 2, 3. Далее следует этап 3 при $h = 2$.

Этап 1 (2) состоит из одного шага и заканчивается случаем 1А (переход ко 2-му этапу).

Этап 2 в этой итерации чрезвычайно прост: цепочка состоит всего из одного элемента. Переходим от X_0 к X_1

$$X_0 = \begin{array}{c} \begin{array}{c} \uparrow \\ \left\| \begin{array}{cccc} 3 & 0 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right\| \begin{array}{c} 1 \\ 0 \\ 0 \\ 2 \end{array} \end{array} \xrightarrow{\theta_1=1} X_1 = \begin{array}{c} \left\| \begin{array}{cccc} 3 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \right\| \begin{array}{c} 0 \\ 0 \\ 0 \\ 2 \end{array} \\ \hline \begin{array}{cccc} 0 & 1 & 2 & 0 \\ 0 & 0 & 2 & 0 \end{array} \end{array}$$

Итак, итерация I состоит из этапов 1, 3, 1 и 2

$$\frac{1}{2} \Delta_1 = 2.$$

Итерация II. Эта итерация проводится без дополнительных пояснений, так как все необходимые замечания были сделаны ранее.

$$C_1 = \begin{array}{c} \oplus \quad \oplus \quad + \\ \left\| \begin{array}{cccc} \bar{0}^* & \bar{0}' & 6 & 3 \\ 8 & \bar{0}' & 4 & 8 \\ 7 & \bar{0}^* & 0' & 4 \\ 3 & 4 & 4 & \bar{0} \end{array} \right\| \begin{array}{c} + \\ + \\ + \\ + \end{array} \end{array} \xrightarrow{h=3} C_1^{(1)} = \begin{array}{c} \begin{array}{cccc} \bar{0}^* & \bar{0}' & 6 & 6 \\ 8 & \bar{0}' & 4 & 11 \\ 7 & \bar{0}^* & 0' & 7 \\ 0 & 1 & 1 & \bar{0} \end{array} \begin{array}{c} + \\ + \\ + \\ + \end{array} \end{array} = C_2$$

$$X_1 = \begin{array}{c} \begin{array}{cccc} 3 & 1 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 3 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 2 \end{array} \end{array} \xrightarrow{\theta_2=2} X_2 = \begin{array}{c} \begin{array}{cccc} 1 & 3 & 0 & 0 \\ 0 & 2 & 0 & 0 \\ 0 & 1 & 2 & 0 \\ 2 & 0 & 0 & 1 \end{array} \begin{array}{c} 0 \\ 0 \\ 0 \\ 0 \end{array} \end{array}$$

Итерация II состоит из этапов 1, 3, 1 и 2. Поскольку $\Delta_2=0$, X_2 является оптимальным планом исследуемой задачи.

3.8. Особенности венгерского метода. Сделаем в заключение несколько замечаний, относящихся к общей характеристике рассмотренного метода. Венгерский метод наиболее эффективен в применении к транспортным задачам с целочисленными объемами производства и потребления. В этом случае число операций, необходимых для решения задачи, может быть оценено заранее. Действительно, если в предварительном этапе алгоритма построена матрица X_0 с суммой невязок Δ_0 , то для получения решения задачи необходимо провести не более $\frac{1}{2} \Delta_0$ итераций (в каждой итерации сумма невязок уменьшается в худшем случае на 2). В большинстве случаев решение транспортной задачи связано с меньшим, чем $\frac{1}{2} \Delta_0$, числом итераций. В рассмотренном только что примере $\frac{1}{2} \Delta_0=3$, а решение было получено за 2 итерации. Что касается отдельной итерации, то ее трудоемкость может быть оценена в зависимости от чисел n и m (проводить соответствующие оценки мы не будем).

Немаловажное достоинство венгерского метода состоит в том, что с его помощью можно не только получить решение задачи транспортировки, но и оценить близость результата каждой из итераций к оптимальному плану перевозок. Имея подобные оценки, можно контролировать ход вычислений и оборвать процесс, как только отклонение от оптимального плана достигнет определенного предела. Это обстоятельство особенно важно для задач с большим числом пунктов производства и потребления, так как часто позволяет существенно сократить вычислительную работу. Остановимся на способе получения указанных оценок.

Пусть результатом k -й итерации венгерского метода, примененного к задаче T , являются матрицы

$$X_k = \|x_{ij}^{(k)}\|; \quad C_k = \|c_{ij}^{(k)}\|.$$

Согласно алгоритму венгерского метода

$$C_k = \|c_{ij}^{(k)}\| = \|c_{ij} - (v_j^{(k)} + u_i^{(k)})\|,$$

где $C = \|c_{ij}\|$ — матрица транспортных издержек задачи T , причем $c_{ij}^{(k)} \geq 0$ для $i = 1, 2, \dots, m$; $j = 1, 2, \dots, n$. Величины $u_i^{(k)}$, $v_j^{(k)}$ нетрудно получить, пользуясь равенством

$$C - C_k = \|c_{ij} - c_{ij}^{(k)}\| = \|v_j^{(k)} + u_i^{(k)}\|.$$

Положим $u_1^{(k)} = 0$, тогда

$$v_j^{(k)} = c_{1j} - c_{1j}^{(k)}, \quad j = 1, 2, \dots, n,$$

$$u_i^{(1)} = c_{i1} - c_{i1}^{(k)} - v_1^{(k)} = c_{i1} - c_{i1}^{(k)} + c_{11}^{(k)} - c_{11}, \quad i = 2, 3, \dots, m.$$

Таким образом, для отыскания $u_1^{(k)}$, $u_2^{(k)}, \dots, u_m^{(k)}$; $v_1^{(k)}$, $v_2^{(k)}, \dots, v_n^{(k)}$ достаточно подсчитать элементы первой строки и первого столбца матрицы

$$C - C_k = \|c_{ij} - c_{ij}^{(k)}\|.$$

Сформулируем, руководствуясь правилами, изложенными в § 1 гл. 4, задачу T^* , сопряженную транспортной задаче T .

Обратить в максимум линейную форму

$$\sum_{j=1}^n b_j v_j + \sum_{i=1}^m a_i u_i \quad (5.48)$$

при соблюдении условий:

$$v_j + u_i \leq c_{ij}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n. \quad (5.49)$$

Поскольку

$$c_{ij}^{(k)} = c_{ij} - (u_i^{(k)} + v_j^{(k)}) \geq 0 \quad \text{для } i = 1, 2, \dots, m, \\ j = 1, 2, \dots, n,$$

то $(u_1^{(k)}, u_2^{(k)}, \dots, u_m^{(k)}; v_1^{(k)}, v_2^{(k)}, \dots, v_n^{(k)})$ — план задачи T^* .

Если через $\mathcal{L}_{\min}$ обозначить минимальные суммарные транспортные издержки в задаче T , то в соответствии со следствием 1 первой теоремы двойственности

$$\sum_{j=1}^n v_j^{(k)} b_j + \sum_{i=1}^m u_i^{(k)} a_i \leq L_{\min}. \quad (5.50)$$

Неравенство (5.50) дает оценку снизу для суммарных транспортных издержек, отвечающих оптимальному плану перевозок. Для получения верхней оценки для $L_{\min}$ используем матрицу $X_k = \|x_{ij}^{(k)}\|$. Вспомним, что при любом k план перевозок $\|x_{ij}^{(k)}\|$ обладает свойствами

$$\begin{aligned} \sum_{i=1}^m x_{ij}^{(k)} &= b_j^{(k)} \leq b_j, \quad j = 1, 2, \dots, n; \\ \sum_{j=1}^n x_{ij}^{(k)} &= a_i^{(k)} \leq a_i, \quad i = 1, 2, \dots, m. \end{aligned}$$

Откуда, положив

$$\Delta b_j^{(k)} = b_j - b_j^{(k)}, \quad \Delta a_i^{(k)} = a_i - a_i^{(k)},$$

получаем

$$\Delta b_j^{(k)} \geq 0, \quad \Delta a_i^{(k)} \geq 0.$$

Выделим из пунктов производства задачи T такие, из которых по плану перевозок X_k вывозится не весь продукт,

$$A_{i_1}, A_{i_2}, \dots, A_{i_s} \quad (\Delta a_{i_\lambda}^{(k)} > 0, \quad \lambda = 1, 2, \dots, s).$$

Аналогично, из пунктов потребления задачи выделим те, которые планом X_k полностью не удовлетворены,

$$B_{j_1}, B_{j_2}, \dots, B_{j_t} \quad (\Delta b_{j_\lambda}^{(k)} > 0, \quad \lambda = 1, 2, \dots, t).$$

Выберем теперь некоторый план перевозок для транспортной задачи $T^{(k)}$ с пунктами производства $A_{i_1}, A_{i_2}, \dots, A_{i_s}$, обладающими объемами $\Delta a_{i_1}^{(k)}, \Delta a_{i_2}^{(k)}, \dots, \Delta a_{i_s}^{(k)}$, и пунктами потребления $B_{j_1}, B_{j_2}, \dots, B_{j_t}$, которым соответствуют объемы потребления $\Delta b_{j_1}^{(k)}, \Delta b_{j_2}^{(k)}, \dots, \Delta b_{j_t}^{(k)}$. Это можно сделать, используя либо метод северо-западного угла (п. 1.3), либо метод минимального эле-

мента (п. 2.4). Обозначим найденный план, дополненный нулевыми перевозками, через

$$\Delta X_k = \|\Delta x_{ij}^{(k)}\|$$

($\Delta x_{ij}^{(k)} = 0$, если либо $i \neq i_\lambda$, $\lambda = 1, 2, \dots, s$, либо $j \neq j_\lambda$, $\lambda = 1, 2, \dots, t$).

Нетрудно убедиться, что

$$\|\tilde{x}_{ij}^{(k)}\| = \|x_{ij}^{(k)}\| + \|\Delta x_{ij}^{(k)}\|$$

— план задачи T . Поэтому

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} \tilde{x}_{ij}^{(k)} \geq L_{\min}. \quad (5.51)$$

Итак, имея матрицы X_k , C_k , являющиеся результатом k -й итерации венгерского метода, можно получить план исследуемой задачи $\tilde{X}_k = \|\tilde{x}_{ij}^{(k)}\|$ и оценить его отклонение от оптимального плана $X^* = \|x_{ij}^*\|$:

$$\begin{aligned} 0 &\leq \sum_{i=1}^m \sum_{j=1}^n c_{ij} \tilde{x}_{ij}^{(k)} - \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^* \leq \\ &\leq \sum_{i=1}^m \sum_{j=1}^n c_{ij} \tilde{x}_{ij} - \left[\sum_{i=1}^m a_i u_i^{(k)} + \sum_{j=1}^n b_j v_j^{(k)} \right] \end{aligned} \quad (5.52)$$

(неравенства (5.52) являются прямым следствием оценок (5.50), (5.51), поскольку $L_{\min} = \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}^*$).

Другим преимуществом венгерского метода является его нечувствительность к явлению вырождения. Существенное достоинство метода состоит также в экономном образовании цепочек (поиск элементов цепочки достаточно прост и не требует предварительного построения разветвленной сети из положительных элементов плана, необходимой в методе потенциалов). Последнее обстоятельство облегчает использование венгерского метода для решения транспортных задач на универсальных математических машинах.

ГЛАВА 6

МАТЕМАТИЧЕСКИЕ ОСНОВЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

В этой главе излагаются некоторые факты из теории выпуклых множеств конечномерных эвклидовых пространств, являющиеся математическим фундаментом линейного программирования. Затем приводятся доказательства обеих теорем двойственности (гл. 4). Здесь же дается обоснование ряда утверждений, касающихся транспортной задачи (гл. 5).

§ 1. КОНЕЧНОМЕРНЫЕ ПРОСТРАНСТВА И ВЫПУКЛЫЕ МНОЖЕСТВА

1.1. Конечномерное векторное пространство*. Упорядоченную систему действительных чисел

$$A = (a_1, a_2, \dots, a_{n-1}, a_n)$$

называют n -мерным вектором с компонентами (координатами) $a_1, a_2, \dots, a_{n-1}, a_n$. Векторы большей частью будут обозначаться заглавной буквой, их компоненты — соответствующими малыми буквами с индексами, указывающими на порядковый номер компоненты.

Каждой паре n -мерных векторов $A = (a_1, a_2, \dots, a_n)$, $B = (b_1, b_2, \dots, b_n)$ сопоставляется вектор $A + B = (a_1 + b_1, a_2 + b_2, \dots, a_n + b_n)$, называемый их суммой. Разностью векторов A, B называется вектор $A - B = (a_1 - b_1, a_2 - b_2, \dots, a_n - b_n)$. Вектор с ну-

* Более подробно с многомерными векторными пространствами читатель может ознакомиться, например, по книге [18].

левыми компонентами $0 = (0, 0, \dots, 0)$ носит название нулевого.

Векторы A, B считаются равными в том и только в том случае, если $A - B = 0$. Произведением вектора A на действительное число α (скаляр) назовем вектор

$$\alpha A = (\alpha a_1, \alpha a_2, \dots, \alpha a_n).$$

Совокупность всех n -мерных векторов, для которых введены понятия сложения, вычитания и умножения на скаляр, называется n -мерным векторным пространством.

При $n = 3$ введенное векторное пространство соответствует обычному пространству трех измерений. Каждой упорядоченной тройке чисел (x, y, z) в этом пространстве соответствует точка с координатами x, y, z (или вектор, проведенный в нее из начала координат). Наоборот, каждой точке 3-мерного пространства может быть сопоставлена упорядоченная тройка чисел, состоящая из ее координат. Таким образом, элементы n -мерного векторного пространства при $n = 3$ (или $n = 2$) можно интерпретировать либо как точки, либо как векторы, проведенные из начала координат. Элементы векторного пространства мы будем в дальнейшем именовать как векторами, так и точками этого пространства.

Заметим, что понятия суммы двух n -мерных векторов и произведения n -мерного вектора на скаляр при $n = 2, 3$ соответствуют хорошо известным правилам обращения с векторами на плоскости и в пространстве.

Вектор $\sum_{i=1}^s \alpha_i A_i$, где A_i — вектор, α_i — скаляр, $i = 1, 2, \dots, s$, называют часто линейной комбинацией векторов $A_1, A_2, \dots, A_s$ с коэффициентами $\alpha_1, \alpha_2, \dots, \alpha_s$.

Если нулевой вектор $0 = (0, 0, \dots, 0)$ может быть представлен в виде линейной комбинации векторов $A_1, A_2, \dots, A_s$, среди коэффициентов которой имеются ненулевые, система векторов $A_1, A_2, \dots, A_s$ называется линейно зависимой. В противном случае система векторов $A_1, A_2, \dots, A_s$ по определению линейно независима. Другими словами, систему векторов $A_1, A_2,$

..., A_s называют линейно независимой, если соотношение

$$\sum_{i=1}^s \alpha_i A_i = 0$$

возможно лишь при $\alpha_1 = \alpha_2 = \dots = \alpha_s = 0$.

Понятие линейной независимости является одним из основных в теории векторных пространств. Приведем без доказательства критерий линейной независимости системы n -мерных векторов, широко используемый в различных разделах математики.

Для линейной независимости системы векторов $A_i = (a_{i1}, a_{i2}, \dots, a_{in})$, $i = 1, 2, \dots, s$ необходимо и достаточно существование квадратной матрицы порядка s , составленной из координат A_i ,

$$\begin{vmatrix} a_{1j_1} & a_{1j_2} & \dots & a_{1j_s} \\ a_{2j_1} & a_{2j_2} & \dots & a_{2j_s} \\ \cdot & \cdot & \cdot & \cdot \\ a_{sj_1} & a_{sj_2} & \dots & a_{sj_s} \end{vmatrix}$$

с определителем, отличным от нуля. Отсюда, в частности, следует, что в n -мерном пространстве не существует линейно независимой системы, состоящей более чем из n векторов. С другой стороны, единичные векторы $e_1 = (1, 0, \dots, 0)$, $e_2 = (0, 1, 0, \dots, 0)$, ..., $e_n = (0, 0, \dots, 1)$ составляют, очевидно, линейно независимую систему.

По аналогии с плоскостью и трехмерным пространством размерностью векторного пространства естественно назвать величину, совпадающую с максимальным числом линейно независимых векторов, в нем содержащихся.

Таким образом, размерность n -мерного векторного пространства в точности равна n . Это обстоятельство послужило поводом для названия данного пространства n -мерным.

Каждой паре векторов n -мерного пространства $A = (a_1, a_2, \dots, a_n)$ и $B = (b_1, b_2, \dots, b_n)$ сопоставляется число

$$(A, B) = \sum_{i=1}^n a_i b_i,$$

называемое их скалярным произведением. Введенная функция пары векторов обладает рядом свойств:

$$\text{а) } (A, B) = (B, A); \quad (6.1)$$

$$\text{б) } (A_1 + A_2, B) = (A_1, B) + (A_2, B); \quad (6.2)$$

$$\text{в) } (\lambda A_1, B) = \lambda (A_1, B); \quad (6.3)$$

$$\text{г) } (A, A) \geq 0, \text{ причем знак равенства возможен лишь при } A = 0; \quad (6.4)$$

$$\text{д) } (A, B)^2 \leq (A, A) \cdot (B, B). \quad (6.5)$$

Соотношения а), б), в), г) проверяются без труда. Докажем справедливость неравенства д), известного под названием неравенства Буняковского.

Введем в рассмотрение функцию

$$\varphi(\lambda) = (A + \lambda B, A + \lambda B).$$

Используя свойства скалярного произведения а), б), в), имеем

$$\varphi(\lambda) = \lambda^2 (B, B) + 2\lambda (A, B) + (A, A).$$

Итак, $\varphi(\lambda)$ — квадратный трехчлен.

Согласно свойству г) $\varphi(\lambda) \geq 0$, $-\infty < \lambda < \infty$, что возможно лишь при неположительности дискриминанта рассматриваемого квадратного трехчлена

$$(A, B)^2 - (A, A)(B, B) \leq 0.$$

Неравенство Буняковского доказано.

Два вектора n -мерного пространства называются ортогональными, если $(A, B) = 0$. В частности, нулевой вектор ортогонален любому вектору данного пространства.

Используя понятие скалярного произведения, можно определить расстояние между точками n -мерного пространства (задать в нем метрику). Длиной или нормой вектора $A = (a_1, \dots, a_n)$ называется число

$$|A| = \sqrt{(A, A)} = \sqrt{\sum_{i=1}^n a_i^2}.$$

Расстояние между точками A, B полагается равным

$$\begin{aligned} \rho(A, B) &= |A - B| = \sqrt{(A - B, A - B)} = \\ &= \sqrt{\sum_{i=1}^n (a_i - b_i)^2}. \end{aligned}$$

В соответствии со свойством г) скалярного произведения, $|A| \geq 0$ для любого вектора A , причем равенство $|A| = 0$ возможно лишь для $A = 0$.

Рассмотрим три произвольные точки n -мерного пространства $A = (a_1, a_2, \dots, a_n)$, $B = (b_1, b_2, \dots, b_n)$, $C = (c_1, c_2, \dots, c_n)$. Покажем, что расстояния между ними связаны неравенством (треугольника):

$$|A - B| \leq |A - C| + |C - B|. \quad (6.6)$$

Положим $A - C = X$; $C - B = Y$. Тогда (6.6) переписывается в виде

$$|X + Y| \leq |X| + |Y|. \quad (6.7)$$

Используя свойства скалярного произведения, имеем

$$|X + Y|^2 = (X + Y, X + Y) = |X|^2 + |Y|^2 + 2(X, Y).$$

Но в соответствии с (6.5) $(X, Y) \leq |X| \cdot |Y|$. Следовательно,

$$|X + Y|^2 \leq |X|^2 + |Y|^2 + 2|X| \cdot |Y| = (|X| + |Y|)^2,$$

что эквивалентно (6.7).

Неравенство треугольника, так же, как и неравенство Буняковского, имеет в случае $n=2, 3$ прозрачный геометрический смысл. Известно, что скалярное произведение двух векторов на плоскости или в пространстве (трехмерном) совпадает с произведением их длин, умноженным на косинус угла между ними. Следовательно, неравенство Буняковского (6.5), будучи переписано в эквивалентной форме $\frac{(A, B)}{|A| \cdot |B|} \leq 1$, означает в данном случае, что косинус любого угла не превышает 1. Что касается неравенства треугольника, то оно расшифровывается так: сумма длин двух сторон треугольника не меньше длины третьей стороны.

n -мерное векторное пространство, в котором определено понятие скалярного произведения (а следовательно, введена метрика) принято называть эвклидовым. В дальнейшем это пространство будет обозначаться через E_n .

Будем говорить, что последовательность векторов $X_1, X_2, \dots, X_k, \dots$ из E_n сходится к вектору $X \in E_n$ ($X_k \rightarrow X$ при $k \rightarrow \infty$), если для любого $\varepsilon > 0$ существует натуральное число $N(\varepsilon)$, начиная с которого ($i \geq N(\varepsilon)$) $|X - X_i| < \varepsilon$.

Из определения нормы в E_n следует, что последовательность $\{X_k\}$ сходится к X в том и только в том случае, если

$$\lim_{k \rightarrow \infty} x_s^{(k)} = x_s, \quad s = 1, 2, \dots, n.$$

Здесь

$$X_k = (x_1^{(k)}, x_2^{(k)}, \dots, x_n^{(k)}), \quad k = 1, 2, \dots,$$

$$X = (x_1, x_2, \dots, x_n).$$

Пользуясь неравенством Буняковского, нетрудно показать, что если

$$X_k \rightarrow X, \quad Y_k \rightarrow Y \quad (\text{при } k \rightarrow \infty), \text{ то}$$

$$\lim_{k \rightarrow \infty} (X_k, Y_k) = (X, Y). \quad (6.8)$$

Мы будем иметь дело с различными множествами точек пространства E_n . Заметим, что условие принадлежности всех точек множества G_1 множеству G_2 записывается обычно так: $G_1 \subset G_2$ (G_1 содержится в G_2). В частности, запись $G_1 \subset E_n$ означает, что G_1 состоит из точек евклидова пространства E_n .

Множество точек $X \in E_n$, удовлетворяющих условию:

$$(\Lambda, X) = \sum_{i=1}^n \lambda_i x_i = c, \quad (6.9)$$

где $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n) \in E_n$, c — действительное число — называется гиперплоскостью с направляющим вектором Λ . Соотношение (6.9), связывающее координаты точек гиперплоскости, является ее уравнением. Из (6.9) вытекает, что вектор Λ ортогонален каждому из векторов $X' - X''$, где X', X'' принадлежат рассматриваемой гиперплоскости. Любая гиперплоскость $[(\Lambda, X) = c]$ порождает пару множеств E_n , называемых полупространствами

$$(\Lambda, X) \geq c; \quad (\Lambda, X) \leq c. \quad (6.10)$$

Множество точек $X \in E_n$, каждая из которых имеет представление

$$X = A + Bt, \quad (6.11)$$

где $A, B \in E_n$, $-\infty < t < \infty$, называется прямой с уравнением (6.11).

Если в (6.11) параметр t ограничен снизу или сверху конечным числом, соответствующее множество называется лучом (или полупрямой). При наличии ограничений на t (конечных) как сверху, так и снизу получаем отрезок. Любую точку отрезка, определяемого соотношением $X = A + Bt$, $\alpha \leq t \leq \beta$, можно представить в виде линейной комбинации его концов $A + B\alpha$; $A + B\beta$. Действительно,

$$A + Bt = \mu_1 (A + \alpha B) + \mu_2 (A + \beta B),$$

где

$$\mu_1 = \frac{\beta - t}{\beta - \alpha}; \quad \mu_2 = \frac{t - \alpha}{\beta - \alpha}.$$

Заметим, что

$$\mu_1 + \mu_2 = 1; \quad \mu_1 \geq 0; \quad \mu_2 \geq 0.$$

Таким образом, отрезок с концами в точках A' , A'' является полной совокупностью точек E_n вида

$$\mu A' + (1 - \mu) A'', \quad (6.12)$$

где $0 \leq \mu \leq 1$.

Сферой с центром в точке $A \in E_n$ и радиусом $r > 0$ называется множество точек $X \in E_n$, для которых

$$|X - A| < r. \quad (6.13)$$

Сфера с центром в точке A радиуса r составляет r -окрестность этой точки.

Если множество $G \subset E_n$ содержит вместе с точкой X ее ε -окрестность при некотором $\varepsilon > 0$, то X называется внутренней точкой G . Внешняя точка множества G обладает некоторой окрестностью, находящейся вне G . Если в любой окрестности точки X содержатся как точки G , так и точки, не принадлежащие G , то X — граничная точка множества G . Множество, содержащее все свои граничные точки, называется замкнутым. Нетрудно убедиться, что гиперплоскость, полупростран-

ство, прямая, луч, отрезок являются замкнутыми множествами. Проверим это утверждение, например, для полупространства H

$$(\Lambda, X) \leq C.$$

Пусть X_0 — произвольная граничная точка H . Тогда существует последовательность точек $H \setminus \{X_k\}$, сходящаяся к X_0 , $\lim_{k \rightarrow \infty} |X_0 - X_k| = 0$. Следовательно, согласно (6.8)

$$(\Lambda, X_0) = \lim_{k \rightarrow \infty} (\Lambda, X_k) \leq C,$$

что доказывает замкнутость H .

Множество G называют *ограниченным*, если найдется такое число c , не зависящее от X , что $|X| < c$ для всех $X \in G$.

На n -мерное эвклидово пространство без труда переносится известная теорема Больцано—Вейерштрасса: *из любой ограниченной последовательности точек E_n может быть выделена сходящаяся подпоследовательность*.

1.2. Выпуклые множества. Множество $G \subset E_n$, содержащее вместе с любыми двумя своими точками A, B соединяющий их отрезок $\mu A + (1 - \mu) B$, $0 \leq \mu \leq 1$, называется *выпуклым*.

Все введенные в предыдущем пункте множества (гиперплоскость, сфера и т. д.) являются выпуклыми.

Читателю будет полезно убедиться в этом самом. Нетрудно установить, что общая часть (пересечение) произвольного числа выпуклых множеств $G_1, G_2, \dots$, обозначаемая обычно ΠG_i , является также выпуклым множеством (читателю рекомендуется это доказать). При исследованиях, связанных с выпуклыми множествами, часто используется следующее важное утверждение.

Теорема о разделяющей гиперплоскости. Пусть G_1, G_2 — произвольные выпуклые замкнутые множества без общих точек, из которых хотя бы одно ограничено. В этих предположениях существует такая гиперплоскость Π , определяемая уравнением

$$(\Lambda, X) = c, \tag{6.14}$$

что $(\Lambda, X) > c$ для всех точек $X \in G_1$, $(\Lambda, X) < c$ для всех точек $X \in G_2$.

Таким образом, согласно сформулированной теореме G_1 принадлежит внутренности одного из полупространств, порождаемых Π , а G_2 — внутренности другого полупространства. Гиперплоскость Π естественно поэтому назвать разделяющей множества G_1 и G_2 .

Доказательство. Для определенности допустим, что ограниченным множеством является G_1 . Рассмотрим

$$\sigma = \inf |P_1 - P_2|,$$

где нижняя грань берется по всевозможным точкам $P_1 \in G_1$ и $P_2 \in G_2$. Согласно определению нижней грани существуют такие последовательности точек $\{P_1^{(k)}\}_k$, $\{P_2^{(k)}\}_k$, $P_1^{(k)} \in G_1$, $P_2^{(k)} \in G_2$, что

$$\lim_{k \rightarrow \infty} |P_1^{(k)} - P_2^{(k)}| = \sigma. \quad (6.15)$$

Поскольку G_1 — ограниченное множество, последовательность $\{P_1^{(k)}\}$ ограничена. Последовательность $\{P_2^{(k)}\}$ также ограничена, так как

$$|P_2^{(k)}| = |(P_2^{(k)} - P_1^{(k)}) + P_1^{(k)}| \leq |P_1^{(k)} - P_2^{(k)}| + |P_1^{(k)}|.$$

Пользуясь теоремой Больцано — Вейерштрасса, выделим из последовательностей $\{P_1^{(k)}\}$, $\{P_2^{(k)}\}$ подпоследовательности, сходящиеся соответственно к точкам P_1^* , P_2^* . В силу замкнутости рассматриваемых множеств $P_1^* \in G_1$, $P_2^* \in G_2$. Таким образом, существует такая последовательность индексов k_i , $i = 1, 2, \dots$, что

$$\lim_{i \rightarrow \infty} |P_1^{(k_i)} - P_1^*| = 0; \quad \lim_{i \rightarrow \infty} |P_2^{(k_i)} - P_2^*| = 0.$$

Переходя в выражении $|P_1^{(k_i)} - P_2^{(k_i)}|$ к пределу при $i \rightarrow \infty$ и учитывая (6.15), получаем

$$|P_1^* - P_2^*| = \sigma = \min_{\substack{P_1 \in G_1 \\ P_2 \in G_2}} |P_1 - P_2|. \quad (6.16)$$

По условию множества G_1 , G_2 не имеют общих точек, следовательно, $\sigma > 0$.

Пусть

$$R_\mu = \mu P_1^* + (1 - \mu) P_2^*,$$

где $0 < \mu < 1$. Обозначим через Π_μ гиперплоскость, определяемую уравнением

$$(\Lambda, X) = (\Lambda, R_\mu), \quad (6.17)$$

где $\Lambda = P_1^* - P_2^*$.

Непосредственной подстановкой убеждаемся, что Π_μ содержит точку R_μ . Покажем, что гиперплоскость Π_μ разделяет G_1 и G_2 . Доказательство проводится от противного.

Пусть $Q \in G_1$ и вместе с тем

$$(\Lambda, Q) \leq (\Lambda, R_\mu). \quad (6.18)$$

Поскольку P_1^* и Q принадлежат выпуклому множеству G_1 , любая точка $Q_\epsilon = (1 - \epsilon)P_1^* + \epsilon Q$, где $0 \leq \epsilon \leq 1$, также содержится в G_1 .

Вычислим квадрат расстояния между точками $Q_\epsilon \in G_1$ и $P_2^* \in G_2$,

$$\begin{aligned} |Q_\epsilon - P_2^*|^2 &= |(1 - \epsilon)P_1^* + \epsilon Q - P_2^*|^2 = \\ &= |\Lambda + \epsilon(Q - P_1^*)|^2 = (\Lambda + \epsilon(Q - P_1^*), \end{aligned}$$

$$\Lambda + \epsilon(Q - P_1^*)) = \epsilon^2 |Q - P_1^*|^2 + 2\epsilon(\Lambda, Q - P_1^*) + |\Lambda|^2.$$

Установим теперь, что

$$(\Lambda, Q - P_1^*) < 0. \quad (6.19)$$

В самом деле,

$$\begin{aligned} (\Lambda, R_\mu) &= (\Lambda, \mu P_1^* + (1 - \mu)P_2^*) = \\ &= (\Lambda, (\mu - 1)(P_1^* - P_2^*) + P_1^*) = \\ &= (\Lambda, (\mu - 1)\Lambda + P_1^*) = (\mu - 1)|\Lambda|^2 + (\Lambda, P_1^*). \end{aligned}$$

Но

$$|\Lambda|^2 = \sigma^2 > 0.$$

Следовательно,

$$(\Lambda, R_\mu) < (\Lambda, P_1^*).$$

Сравнивая полученное соотношение с (6.18), приходим к неравенству:

$$(\Lambda, P_1^*) > (\Lambda, Q),$$

эквивалентному (6.19). Обозначим

$$(\Lambda, Q - P_1^*) = \alpha \ (\alpha < 0), \quad |Q - P_1^*| = \beta.$$

Тогда

$$|Q_\epsilon - P_2^*|^2 = \beta^2 \epsilon^2 + 2\epsilon\alpha + \sigma^2.$$

Выбрав $\theta > 0$ из условий: — $\frac{\alpha}{\beta^2} \theta < 1^*$, $\theta \leq 1$, положим $\varepsilon_0 = -\frac{\alpha}{\beta^2} \theta$. Так как $0 < \varepsilon_0 < 1$, $Q_{\varepsilon_0} \in G_1$. С другой стороны,

$$|Q_{\varepsilon_0} - P_2^*| = \sigma^2 + 2\varepsilon_0\alpha + \varepsilon_0^2\beta^2 = \sigma^2 - \frac{\alpha^2}{\beta^2} \theta (2 - \theta) < \sigma^2.$$

Итак, допустив справедливость неравенства (6.18), приходим к соотношению, противоречащему (6.16). Следовательно, $(\Lambda, Q) > (\Lambda, R_\mu)$ для любой точки $Q \in G_1$.

Аналогично доказывается, что $(\Lambda, Q) < (\Lambda, R_\mu)$ для всех точек $Q \in G_2$.

Таким образом, гиперплоскость

$$(\Lambda, X) = c, \text{ где } \Lambda = P_1^* - P_2^*, \quad c = (\Lambda, R_\mu),$$

$$R_\mu = \mu P_1^* + (1 - \mu) P_2^*, \quad 0 < \mu < 1$$

разделяет множества G_1 и G_2 . Теорема полностью доказана.

Из только что установленной теоремы можно извлечь несколько полезных следствий.

Следствие 1. Пусть P_0 — граничная точка выпуклого замкнутого множества G . Существует такая гиперплоскость $\Pi: (\Lambda, X) = c$, что

$$а) (\Lambda, P_0) = c \text{ (}\Pi \text{ содержит точку } P_0\text{);}$$

$$б) (\Lambda, P) \leq c, \quad P \in G$$

(G лежит в одном из полупространств, порождаемых Π). Гиперплоскость, удовлетворяющая условиям а), б), называется опорной гиперплоскостью множества G в точке P_0 .

Доказательство. Поскольку P_0 — граничная точка G , найдется последовательность $\{P_k\}$, состоящая из внешних точек множества G и сходящаяся к P_0 . Рассмотрим выпуклые множества G_1 и G_2 , из которых первое состоит лишь из одной точки P_k , а второе совпадает с G .

) Предполагается, что $|Q - P_1^| = \beta > 0$. Если $\beta = 0$, $|Q_\varepsilon - P_2^*|^2 = \sigma^2 + 2\varepsilon\alpha < \sigma^2$ при любом $\varepsilon > 0$.

По теореме о разделяющей гиперплоскости существует такая гиперплоскость Π_k :

$$(\Lambda_k, X) = c_k, \quad (6.20)$$

что

$$(\Lambda_k, P_k) > c_k; \quad (6.21)$$

$$(\Lambda_k, P) < c_k \text{ для } P \in G. \quad (6.22)$$

Не уменьшая общности, можно считать $|\Lambda_k| \leq 1$; $|c_k| \leq 1$. Пользуясь теоремой Больцано—Вейерштрасса, выделим из последовательности векторов $\{\Lambda_k\}$ и последовательности чисел $\{c_k\}$ подпоследовательности, сходящиеся соответственно к Λ и c . Покажем, что гиперплоскость $(\Lambda, X) = c$ является искомой. Действительно, переходя при фиксированном $P \in G$ в неравенстве (6.22) к пределу по выделенной подпоследовательности индексов « k », получаем б). В частности,

$$(\Lambda, P_0) \leq c. \quad (6.23)$$

С другой стороны, переход к пределу по этой подпоследовательности в (6.21) приводит к неравенству

$$(\Lambda, P_0) \geq c. \quad (6.24)$$

Сравнивая (6.23) и (6.24), делаем вывод о справедливости условия а). Существование опорной гиперплоскости множества G в точке P_0 доказано.

Определим расстояние между точкой P и множеством G , положив

$$\rho(P, G) = \inf_{Q \in G} |P - Q|.$$

Пусть G_1, G_2 — замкнутые непересекающиеся множества E_n , из которых хотя бы одно ограничено. Назовем $P_2^* \in G_2$ точкой G_2 , наименее удаленной от множества G_1 , если

$$\rho(P_2^*, G_1) = \inf_{P_2 \in G_2} \rho(P_2, G_1).$$

Существование такой точки было установлено при доказательстве теоремы о разделяющей гиперплоскости.

Следствие 2. Пусть G_1, G_2 — множества, удовлетворяющие условиям теоремы о разделяющей гиперплоскости, P_2^* — точка G_2 , наименее удаленная от G_1 .

Существует такая опорная гиперплоскость множества G_2 в точке P_2 :

$$(\Lambda, X) = c, \text{ что } (\Lambda, P_1) > c \text{ для всех } P_1 \in G_1.$$

Доказательство. Введем в рассмотрение гиперплоскость $\Pi_\mu: (\Lambda, X) = (\Lambda, R_\mu)$, где $\Lambda = P_1^* - P_2^*$, $R_\mu = \mu P_1^* + (1 - \mu) P_2^*$, P_1^* — точка G_1 , наименее удаленная от G_2 , $0 < \mu < 1$. В процессе доказательства теоремы было установлено, что Π_μ при $0 < \mu < 1$ разделяет множества G_1 и G_2 , т. е.

$$\min_{P_1 \in G_1} (\Lambda, P_1) > (\Lambda, R_\mu), \quad (6.25)$$

$$\max_{P_2 \in G_2} (\Lambda, P_2) < (\Lambda, R_\mu). \quad (6.26)$$

Устремляя в (6.26) параметр μ к нулю, получаем

$$\max_{P_2 \in G_2} (\Lambda, P_2) \leq (\Lambda, P_2^*). \quad (6.27)$$

Заметим, что (Λ, R_μ) монотонно убывает с убыванием μ . Действительно,

$$(\Lambda, R_\mu) = \mu |\Lambda|^2 + (\Lambda, P_2^*),$$

причем $|\Lambda|^2 = \sigma^2 > 0$. Поэтому, если в (6.25) устремить μ к нулю, то неравенство останется строгим:

$$\min_{P_1 \in G_1} (\Lambda, P_1) > (\Lambda, P_2^*). \quad (6.28)$$

Из соотношений (6.27), (6.28) вытекает, что гиперплоскость Π_0

$$(\Lambda, X) = (\Lambda, P_2^*)$$

обладает всеми требуемыми свойствами и, следовательно, является искомой.

1.3. Крайние точки множества и их свойства. Точку P множества G назовем крайней, если не существует двух различных точек P_1, P_2 , принадлежащих G , таких, что

$$P = \mu P_1 + (1 - \mu) P_2,$$

где $0 < \mu < 1$.

Таким образом, если P — крайняя точка множества G , а S — произвольный отрезок, принадлежащий G , то P либо не содержится в S , либо совпадает с одним из его концов. Примерами крайних точек линейных и плоских множеств являются концы отрезка, вершины многоугольника, точки граничной окружности круга и т. д. Выпуклые множества могут обладать как конечным (многоугольник), так и бесконечным (круг) числом крайних точек. Некоторые выпуклые множества (например, прямая) вовсе не содержат крайних точек. Ниже мы увидим, что подобные множества обязаны быть либо неограниченными (прямая), либо незамкнутыми (открытый интервал). Будем говорить, что множество G является выпуклой оболочкой множества G_1 , если оно состоит из всех возможных точек вида

$$\sum_{k=1}^N \mu_k P_k,$$

где

$$\sum_{k=1}^N \mu_k = 1, \quad \mu_k \geq 0, \quad k = 1, 2, \dots, N;$$

$P_1, P_2, \dots, P_N$ — произвольная конечная система точек G . Вектор $\sum_{k=1}^N \mu_k P_k$ называют обычно выпуклой линейной комбинацией векторов $P_1, P_2, \dots, P_N$. Нетрудно заметить, что выпуклая оболочка произвольного множества есть выпуклое множество. С другой стороны, любое выпуклое множество, содержащее G_1 , обязано включать его выпуклую оболочку G^*). Поэтому выпуклая оболочка G_1 совпадает с минимальным выпуклым множеством, содержащим G_1 .

Теорема о представлении выпуклого множества. Пусть G — выпуклое, замкнутое, ограниченное множество; G^* — совокупность крайних точек G . В таком случае выпуклая оболочка G^* совпадает с G .

Доказательство. Доказательство проводится индукцией по числу измерений « n » евклидова пространства E_n , содержащего G . Если выпуклое множество $G \subset E_1$, то оно, очевидно, совпадает с некоторым

*) Рекомендуем читателю оба эти утверждения доказать самому.

отрезком, любая точка которого является выпуклой комбинацией его концов, составляющих в данном случае множество G^* . Итак, при $n=1$ теорема доказана.

Допустим справедливость утверждения теоремы для пространства E_{n-1} . Покажем, что оно имеет место и в случае E_n . Рассмотрим произвольное выпуклое замкнутое ограниченное множество $G \subset E_n$.

а) Допустим вначале, что $P_0 = (p_1^{(0)}, p_2^{(0)}, \dots, p_n^{(0)})$ — граничная точка G . Тогда по следствию I предыдущего пункта через точку P_0 может быть проведена гиперплоскость Π :

$$(\Lambda, X) = c, \quad (6.29)$$

опорная для множества G . Через G_1 обозначим общую часть (пересечение) множества G и гиперплоскости Π . Очевидно, G_1 — выпуклое, замкнутое, ограниченное множество; $P_0 \in G_1$. Допустим, что отличной от нуля компонентой направляющего вектора $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_n)$ является последняя (разумеется, это допущение ни в коей мере не повлияет на общность наших дальнейших рассуждений). В соответствии с (6.29) для любой точки $P = (p_1, p_2, \dots, p_n) \in \Pi$ имеет место равенство

$$p_n = -\frac{1}{\lambda_n} \sum_{k=1}^{n-1} \lambda_k p_k + \frac{c}{\lambda_n}. \quad (6.30)$$

Рассмотрим множество $\bar{G}_1 \subset E_{n-1}$, состоящее из всех таких точек $\bar{P} = (p_1, p_2, \dots, p_{n-1}) \in E_{n-1}$, что $P = (p_1, p_2, \dots, p_{n-1}, p_n)$, где p_n определяется соотношением (6.30), принадлежит G_1 . $\bar{G}_1$ — выпуклое ограниченное замкнутое множество (этим условиям удовлетворяет множество G_1); $\bar{P}_0 = (p_1^{(0)}, p_2^{(0)}, \dots, p_{n-1}^{(0)}) \in \bar{G}_1$. Следовательно, по предположению индукции найдутся такие крайние точки $\bar{G}_1$: $\bar{P}_1 = (p_1^{(1)}, p_2^{(1)}, \dots, p_{n-1}^{(1)})$, $\bar{P}_2 = (p_1^{(2)}, p_2^{(2)}, \dots, p_{n-1}^{(2)})$, $\dots$, $\bar{P}_N = (p_1^{(N)}, p_2^{(N)}, \dots, p_{n-1}^{(N)})$, что

$$\bar{P}_0 = \sum_{i=1}^N \mu_i \bar{P}_i; \quad (6.31)$$

$$\sum_{i=1}^N \mu_i = 1, \quad \mu_i \geq 0, \quad i = 1, 2, \dots, N.$$

Определим точки $P_i = (p_1^{(i)}, p_2^{(i)}, \dots, p_{n-1}^{(i)}, p_n^{(i)})$, $i = 1, 2, \dots, N$, положив

$$p_n^{(i)} = - \sum_{k=1}^{n-1} \frac{\lambda_k}{\lambda_n} p_k^{(i)} + \frac{c}{\lambda_n}. \quad (6.32)$$

Все точки $P_i \in G_1$, так как по условию $\bar{P}_i \in \bar{G}_1$, $i = 1, 2, \dots, N$. Поскольку точка $P_0 = (p_1^{(0)}, p_2^{(0)}, \dots, p_n^{(0)})$ удовлетворяет соотношению (6.30) и имеет место представление (6.31),

$$\begin{aligned} p_n^{(0)} &= - \sum_{k=1}^{n-1} \frac{\lambda_k}{\lambda_n} p_k^{(0)} + \frac{c}{\lambda_n} = - \sum_{k=1}^{n-1} \frac{\lambda_k}{\lambda_n} \sum_{s=1}^N \mu_s p_k^{(s)} + \frac{c}{\lambda_n} = \\ &= \sum_{s=1}^N \mu_s \left[- \sum_{k=1}^{n-1} \frac{\lambda_k}{\lambda_n} p_k^{(s)} + \frac{c}{\lambda_n} \right] = \sum_{s=1}^N \mu_s p_n^{(s)}. \end{aligned} \quad (6.33)$$

Учитывая (6.31) и (6.33), получаем

$$P_0 = \sum_{i=1}^N \mu_i P_i, \quad (6.34)$$

где $\sum_{i=1}^N \mu_i = 1$, $\mu_i \geq 0$.

Покажем, что все точки, участвующие в представлении (6.34), — крайние точки множества G . Предположив противное для некоторого i , $1 \leq i \leq N$, имеем

$$P_i = \mu Q_1 + (1 - \mu) Q_2; \quad 0 < \mu < 1; \quad Q_1 \neq Q_2;$$

$$Q_1 = (q_1^{(1)}, q_2^{(1)}, \dots, q_n^{(1)}) \in G, \quad Q_2 = (q_1^{(2)},$$

$$q_2^{(2)}, \dots, q_n^{(2)}) \in G.$$

Так как Π — опорная гиперплоскость множества G ,

$$(\Lambda, Q_1) \leq c; \quad (\Lambda, Q_2) \leq c. \quad (6.35)$$

С другой стороны, $P_i \in \Pi$, т. е.

$$(\Lambda, P_i) = c.$$

Откуда

$$\begin{aligned} (\Lambda, Q_1) &= \left(\Lambda, \frac{P_i - (1 - \mu) Q_2}{\mu} \right) = \\ &= \frac{c - (1 - \mu) (\Lambda, Q_2)}{\mu} \geq \frac{c - (1 - \mu) c}{\mu} = c. \end{aligned} \quad (6.36)$$

Аналогично,

$$(\Lambda, Q_2) \geq c. \quad (6.37)$$

Сравнивая (6.35), (6.36) и (6.37), получаем $(\Lambda, Q_1) = (\Lambda, Q_2) = c$, т. е. точки Q_1, Q_2 содержатся как в G , так и в Π . Поэтому $Q_1, Q_2 \in G_1$, а следовательно,

$$\bar{Q}_1 = (q_1^{(1)}, q_2^{(1)}, \dots, q_{n-1}^{(1)}) \in \bar{G}_1;$$

$$\bar{Q}_2 = (q_1^{(2)}, q_2^{(2)}, \dots, q_{n-1}^{(2)}) \in \bar{G}_1.$$

Итак,

$$\bar{P}_i = \mu \bar{Q}_1 + (1 - \mu) \bar{Q}_2,$$

где $\bar{Q}_1, \bar{Q}_2 \in \bar{G}_1$, $0 < \mu < 1$. Поскольку $\bar{P}_i$ — крайняя точка множества $\bar{G}_1$, $\bar{Q}_1 = \bar{Q}_2$. Откуда, обращаясь к соотношению (6.30), выводим, что $Q_1 = Q_2$. Получили противоречие. Следовательно, P_i — крайняя точка множества G .

б) Пусть теперь P_0 — внутренняя точка G .

Рассмотрим прямую L

$$X = P_0 + Rt, \quad (6.38)$$

где R — произвольный ненулевой вектор, $-\infty < t < \infty$. Пересечение G и L совпадает, очевидно, с отрезком $\bar{L}$: $X = P_0 + Rt$, $t_1 \leq t \leq t_2$ (G — ограниченное выпуклое замкнутое множество). Положим

$$X_1 = P_0 + Rt_1; \quad X_2 = P_0 + Rt_2.$$

Тогда

$$P_0 = \mu X_1 + (1 - \mu) X_2,$$

где

$$0 < \mu = \frac{t_2}{t_2 - t_1} < 1 \quad (t_1 < 0, t_2 > 0).$$

Поскольку, как нетрудно заметить, X_1, X_2 — граничные точки G , для них по доказанному существует представление типа (6.34)

$$X_1 = \sum_{i=1}^{N_1} \mu_i^{(1)} P_i^{(1)}; \quad X_2 = \sum_{i=1}^{N_2} \mu_i^{(2)} P_i^{(2)}; \quad \sum_{i=1}^{N_1} \mu_i^{(1)} = 1;$$

$$\mu_i^{(1)} \geq 0; \quad \sum_{i=1}^{N_2} \mu_i^{(2)} = 1; \quad \mu_i^{(2)} \geq 0.$$

Следовательно,

$$P_0 = \sum_{i=1}^{N_1} \mu \mu_i^{(1)} P_i^{(1)} + \sum_{i=1}^{N_2} (1 - \mu) \mu_i^{(2)} P_i^{(2)}.$$

Положим

$$P_i = \begin{cases} P_i^{(1)}, & i = 1, 2, \dots, N_1; \\ P_{i-N_1}^{(2)}, & i = N_1 + 1, N_1 + 2, \dots, N_1 + N_2 = N. \end{cases}$$

$$\mu_i = \begin{cases} \mu \mu_i^{(1)}, & i = 1, 2, \dots, N_1; \\ (1 - \mu) \mu_{i-N_1}^{(2)}, & i = N_1 + 1, N_1 + 2, \dots, N_1 + N_2 = N. \end{cases}$$

Тогда

$$P_0 = \sum_{i=1}^N \mu_i P_i; \quad (6.39)$$

$$\sum_{i=1}^N \mu_i = 1, \quad \mu_i \geq 0, \quad i = 1, 2, \dots, N.$$

Итак, произвольная точка $P_0 \in G$ может быть представлена в виде выпуклой комбинации конечного числа крайних точек этого множества. Вместе с тем любая выпуклая комбинация точек выпуклого множества принадлежит этому множеству.

Таким образом, допустив справедливость утверждения теоремы для эвклидова пространства E_{n-1} размерности $(n-1)$, мы установили, что оно имеет место и для E_n . Вместе с тем, как было показано ранее, теорема верна для случая $n=1$. Следовательно, теорема о представлении выпуклого множества доказана в полном объеме.

Пусть G — произвольное выпуклое множество.

Множество G_0 назовем **остовом** G , если

а) G — выпуклая оболочка G_0 ;

б) любое подмножество G_0 условию а) не удовлетворяет.

Образно говоря, остов множества является наиболее экономным множеством из числа удовлетворяющих условию а). Заметим, что не любое выпуклое множество обладает остовом. Например, открытый интервал остова не имеет. Теорема о представлении устанавливает существование остова для произвольного выпуклого замкнутого ограниченного множества.

При этом показывается, что остов множества G , удовлетворяющего перечисленным условиям, совпадает с G^* — множеством крайних точек G .

Напомним определение выпуклого многогранника, данное в гл. 1.

Выпуклое замкнутое ограниченное множество $G \subset E_n$, обладающее конечным числом крайних точек, называется выпуклым многогранником. Крайние точки многогранника принято называть его вершинами. В качестве непосредственного вывода из теоремы о представлении получаем

Следствие 1. *Любая точка P выпуклого многогранника G с вершинами $P_1, P_2, \dots, P_s$ может быть представлена в виде*

$$P = \sum_{i=1}^s \mu_i P_i; \quad (6.40)$$

$$\sum_{i=1}^s \mu_i = 1, \quad \mu_i \geq 0, \quad i = 1, 2, \dots, s.$$

Приведем еще одно очевидное следствие теоремы о представлении.

Следствие 2. *Всякое непустое выпуклое ограниченное замкнутое множество содержит хотя бы одну крайнюю точку.*

1.4. Выпуклые конусы. В этом пункте будет рассмотрен важный класс выпуклых множеств, называемых конусами.

Выпуклое замкнутое множество $T \subset E_n$ назовем выпуклым конусом с вершиной в точке P_0 , если для произвольного вектора $P \in T$ и любого $\mu \geq 0$ вектор $P_0 + \mu(P - P_0) \in T$. Достаточно общим примером конусов в трехмерном пространстве является следующий. Пусть D — произвольная выпуклая плоская область, замкнутая и ограниченная, P_0 — точка, лежащая вне плоскости, содержащей D . Совокупность всех полупрямых (лучей), исходящих из P_0 и пересекающих D , является выпуклым конусом. Если, например, D — круг и P_0 лежит на перпендикуляре, восстановленном из центра D к плоскости, содержащей этот круг, получаем хорошо известный прямой круговой конус. Подобные построения могут быть проведены и в n -мерном евклидовом пространстве при $n > 3$. Отметим одно свой-

ство выпуклых конусов. Пусть Π — гиперплоскость, опорная для конуса T в точке P . В таком случае эта гиперплоскость содержит любую точку P' вида

$$P' = P_0 + \mu(P - P_0), \quad (6.41)$$

где P_0 — вершина T , $\mu \geq 0$.

Положим, что уравнением Π является

$$(\Lambda, X) = c. \quad (6.42)$$

Допустим для определенности, что величина μ в соотношении (6.41) больше 1. Если P'' — произвольная точка T , имеющая представление (6.41) с $\mu < 1$, то найдется такое число s , $0 < s < 1$, что

$$P = sP' + (1 - s)P''.$$

Подставляя полученное выражение в уравнение (6.42), имеем

$$s[(\Lambda, P') - c] + (1 - s)[(\Lambda, P'') - c] = 0. \quad (6.43)$$

Поскольку Π — опорная гиперплоскость конуса T , каждая из величин, стоящих в квадратных скобках выражения (6.43), неположительна.

Следовательно, для справедливости равенства (6.43) необходимо ($0 < s < 1$), чтобы обе они совпадали с нулем, т. е.

$$(\Lambda, P') = (\Lambda, P'') = c.$$

Итак, произвольная точка P' , лежащая на луче, исходящем из P_0 в сторону P , принадлежит опорной гиперплоскости Π .

В частности, любая опорная гиперплоскость конуса T содержит его вершину.

Введем в рассмотрение множество K_{P_0} , состоящее из всех тех точек P , для которых имеет место следующее представление:

$$P = \sum_{k=1}^N x_k A_k + P_0,$$

где $A_1, A_2, \dots, A_N, P_0$ — точки (вектора) E_n ; $x_1, x_2, \dots, x_N$ — неотрицательные числа. Покажем, что K_{P_0} — выпуклый конус с вершиной в точке P_0 . Пусть $P', P'' \in K_{P_0}$;

$$P' = \sum_{k=1}^N x_k' A_k + P_0; \quad x_k' \geq 0;$$

$$P'' = \sum_{k=1}^N x_k'' A_k + P_0; \quad x_k'' \geq 0.$$

Тогда при любом $0 \leq \mu \leq 1$

$$\begin{aligned} P_\mu - P_0 &= \mu P' + (1 - \mu) P'' - P_0 = \mu (P' - P_0) + \\ &+ (1 - \mu) (P'' - P_0) = \mu \sum_{k=1}^N x_k' A_k + \\ &+ (1 - \mu) \sum_{k=1}^N x_k'' A_k = \sum_{k=1}^N x_k^{(\mu)} A_k, \end{aligned}$$

где

$$x_k^{(\mu)} = \mu x_k' + (1 - \mu) x_k'' \geq \min(x_k', x_k'') \geq 0,$$

$$k = 1, 2, \dots, N.$$

Следовательно, $P_\mu \in K_{P_0}$. Итак, K_{P_0} — выпуклое множество.

Если $P \in K_{P_0}$, $P = \sum_{k=1}^N x_k A_k + P_0$, то $\mu(P - P_0) + P_0$ при любом $\mu \geq 0$ также принадлежит K_{P_0} .

Действительно,

$$\mu(P - P_0) + P_0 = \sum_{k=1}^N (\mu x_k) A_k + P_0.$$

В дальнейшем будет показано, что K_{P_0} — замкнутое множество. Таким образом, множество K_{P_0} действительно является выпуклым конусом с вершиной в точке P_0 . Характерная особенность конуса K_{P_0} состоит в том, что для его образования достаточно иметь лишь конечное число векторов $A_1, A_2, \dots, A_N, P_0$, которые, как говорят, порождают этот конус. Если обратиться к изложенному выше методу построения выпуклых конусов

в трехмерном пространстве, то в данном случае область D является выпуклым многоугольником^{*)}. В пространствах большего числа измерений роль области D в образовании конуса K_{P_0} играет выпуклый многогранник. Поэтому выпуклые конусы, порожденные конечным числом векторов, называются **многогранными**. При доказательстве теорем двойственности (§ 2 настоящей главы) нами будут использованы некоторые свойства многогранных конусов из E_n . Соответствующие утверждения формулируются ниже в виде трех вспомогательных лемм. В дальнейшем через K обозначается выпуклый многогранный конус с вершиной в начале координат, образованный векторами $A_1, A_2, \dots, A_N$.

Лемма 1. *Любая точка $P \in K$ может быть представлена в виде*

$$P = \sum_{k=1}^s x_{ik} A_{ik}, \quad x_{ik} > 0, \quad k = 1, 2, \dots, s, \quad (6.44)$$

где векторы $A_{i_1}, A_{i_2}, \dots, A_{i_s}$ образуют линейно независимую систему.

Доказательство. Рассмотрим представление точки P с минимальным числом положительных коэффициентов x_i . Пусть (6.44) является одним из таких представлений. Допуская, что система векторов $A_{i_1}, A_{i_2}, \dots, A_{i_s}$ линейно зависима, получаем равенство

$$\sum_{k=1}^s t_k A_{i_k} = 0, \quad (6.45)$$

в котором среди $t_k, 1 \leq k \leq s$, имеются отличные от нуля. Пусть для определенности хотя бы одна из величин t_k положительна. Умножая (6.45) на $\theta > 0$ и вычитая результат из (6.44), имеем

$$P = \sum_{k=1}^s (x_{ik} - \theta t_k) A_{i_k}.$$

Если теперь положить $\theta = \min_{t_k > 0} \frac{x_{ik}}{t_k}$, то приходим к новому представлению P , в котором число положительных коэффициентов не превосходит $s - 1 < s$.

^{*)} В общем случае D — многоугольная область (быть может, неограниченная).

Полученное противоречие убеждает нас в линейной независимости системы $A_{i_1}, A_{i_2}, \dots, A_{i_s}$. Лемма доказана.

Через Π_Q обозначим опорную гиперплоскость конуса в точке Q .

Лемма 2. Для произвольной граничной точки P конуса K найдется такое $\varepsilon > 0$, что любая опорная гиперплоскость Π_Q , где $|Q - P| < \varepsilon$, содержит P .

Доказательство. Допустим противное, т. е. существование такой последовательности точек $\{Q_r\}$, что $\lim_{r \rightarrow \infty} |P - Q_r| = 0$, однако ни одна из гиперплоскостей Π_{Q_r} не содержит P . Пусть

$$Q_r = \sum_{k=1}^{s_r} x_{i_k}^{(r)} A_{i_k, r},$$

где $x_{i_k}^{(r)} > 0$, $k = 1, 2, \dots, s_r$. В соответствии с леммой 1 для любого $r = 1, 2, \dots$ систему векторов $A_{i_1, r}, A_{i_2, r}, \dots, A_{i_{s_r}, r}$ можно считать линейно независимой.

Поскольку число точек Q_r бесконечно, а количество всевозможных линейно независимых систем, которые могут быть составлены из векторов $A_1, A_2, \dots, A_N$, заведомо конечно, из последовательности $\{Q_r\}$ выделяется такая бесконечная подпоследовательность $\{Q_{r_t}\}$, что

$$\{Q_{r_t}\} = \sum_{k=1}^s x_{i_k}^{(t)} A_{i_k},$$

где линейно независимая система $\{A_{i_k}\}$ одна и та же для всех точек Q_{r_t} , $x_{i_k}^{(t)} > 0$, $k = 1, 2, \dots, s$, $t = 1, 2, \dots$. Итак, получена бесконечная последовательность точек $\{\bar{Q}_t\}$, обладающая следующими свойствами:

а) $\bar{Q}_t = \sum_{k=1}^s x_{i_k}^{(t)} A_{i_k}$, где векторы $A_{i_1}, A_{i_2}, \dots, A_{i_s}$ линейно независимы, $x_{i_k}^{(t)} > 0$, $k = 1, 2, \dots, s$, $t = 1, 2, \dots$;

б) $\lim_{t \rightarrow \infty} |\bar{Q}_t - P| = 0$;

в) $P \in \Pi_{\bar{Q}_t}$ для любого $t = 1, 2, \dots$.

Из условия б) и линейной независимости векторов $A_{i_1}, A_{i_2}, \dots, A_{i_s}$ можно сделать вывод об ограниченности коэффициентов $x_{i_k}^{(t)}$, $t = 1, 2, \dots$; $k = 1, 2, \dots, s$.

В самом деле, если бы при некотором k из последовательности $\{x_{i_k}^{(t)}\}_t$ можно было бы выделить подпоследовательность $\{\bar{x}_{i_k}^{(t)}\}_t$, сходящуюся к бесконечности, то

$$\lim_{t \rightarrow \infty} \sum_{k=1}^s \bar{x}_{i_k}^{(t)} A_{i_k} = 0,$$

где

$$\bar{x}_{i_k}^{(t)} = \frac{\bar{x}_{i_k}^{(t)}}{\sum_{k=1}^s \bar{x}_{i_k}^{(t)}}.$$

Выделяя из ограниченных последовательностей $\{\bar{x}_{i_k}^{(t)}\}_t$, $k = 1, 2, \dots, s$, подпоследовательности, сходящиеся соответственно к числам $\bar{x}_{i_1}, \bar{x}_{i_2}, \dots, \bar{x}_{i_s}$, получаем

$$\sum_{k=1}^s \bar{x}_{i_k} A_{i_k} = 0. \quad (6.46)$$

Поскольку $\sum_{k=1}^s \bar{x}_{i_k} = 1$, среди величин $\bar{x}_{i_k}$ имеются положительные. Следовательно, равенство (6.46) противоречит линейной независимости векторов $A_{i_1}, A_{i_2}, \dots, A_{i_s}$. Таким образом, каждая из последовательностей $\{x_{i_k}^{(t)}\}_t$, $k = 1, 2, \dots, s$, ограничена. Поэтому существует такая последовательность индексов $t_1, t_2, \dots, t_\alpha, \dots$, что

$$\lim_{\alpha \rightarrow \infty} x_{i_k}^{(t_\alpha)} = x_{i_k}, \quad k = 1, 2, \dots, s. \quad (6.47)$$

Из (6.47) вытекает равенство

$$\lim_{\alpha \rightarrow \infty} \left| \sum_{k=1}^s x_{i_k} A_{i_k} - \bar{Q}_{t_\alpha} \right| = 0.$$

Откуда в соответствии с б)

$$P = \sum_{k=1}^s x_{i_k} A_{i_k}, \quad x_{i_k} \geq 0, \quad k = 1, 2, \dots, s. \quad (6.48)$$

Покажем теперь, что любая из опорных гиперплоскостей $\Pi_{\bar{Q}_{t_\alpha}}$, $\alpha = 1, 2, \dots$ содержит точку P . Пусть уравнением $\Pi_{\bar{Q}_{t_\alpha}}$ является $(\Lambda_\alpha, X) = 0$ (вершина конуса K расположена в начале координат). Тогда

$$(\Lambda_\alpha, \bar{Q}_{t_\alpha}) = \sum_{k=1}^s x_{i_k}^{(t_\alpha)} (\Lambda_\alpha, A_{i_k}) = 0.$$

Поскольку $(\Lambda_\alpha, A_{i_k}) \leq 0$ ($\Pi_{\bar{Q}_{t_\alpha}}$ — опорная гиперплоскость конуса K), а $x_{i_k}^{(t_\alpha)} > 0$,

$$(\Lambda_\alpha, A_{i_k}) = 0, \quad k = 1, 2, \dots, s.$$

Откуда, обращаясь к (6.48), приходим к искомому равенству

$$(\Lambda_\alpha, P) = 0,$$

означающему, что $P \in \Pi_{\bar{Q}_{t_\alpha}}$, $\alpha = 1, 2, \dots$

Выявленное противоречие между условиями а), б), в) служит доказательством справедливости утверждения леммы 2.

Заметим, что в процессе доказательства леммы 2 было, в частности, установлено, что из любой ограниченной последовательности точек K можно выделить подпоследовательность, сходящуюся к некоторой точке рассматриваемого конуса. Следовательно, конус K (а значит и любой конус K_{P_0}) является замкнутым множеством.

Пусть $P \in K$. Через K_P обозначим выпуклый многогранный конус с вершиной в точке P , образованный произвольными векторами $B_1, B_2, \dots, B_{N_1}$ ($B_i \neq 0$).

Будем предполагать, что при любых $y_i \geq 0$. $\left(\sum_{i=1}^{N_1} y_i > 0 \right)$

$$\sum_{i=1}^{N_1} y_i B_i \neq 0^*).$$

*) В этом и только в этом случае вершина конуса K_P будет его крайней точкой.

Лемма 3. Если общая часть выпуклых многогранных конусов K, K_P содержит лишь точку P , то существует гиперплоскость Π

$$(\Lambda, X) = 0;$$

опорная для K в точке P и такая, что

$$(\Lambda, Q) > 0 \quad (6.49)$$

для всех точек $Q \in K_P, Q \neq P$.

Доказательство. Введем в рассмотрение множество $K_P^{(\delta)}$, состоящее из точек

$$Q = \sum_{k=1}^{N_1} y_k B_k + P \in K_P,$$

для которых

$$\delta \leq \sum_{k=1}^{N_1} y_k \leq 1.$$

Нетрудно видеть, что $K_P^{(\delta)}$ — выпуклое замкнутое и ограниченное множество, не имеющее общих точек с K . Через P_δ обозначим точку K , наименее удаленную от $K_P^{(\delta)}$. Очевидно, она является граничной точкой конуса K . Пусть Π_δ — гиперплоскость, опорная для K в точке P_δ и не содержащая $K_P^{(\delta)}$ (ее существование гарантируется следствием 2 теоремы о разделяющей гиперплоскости). Убедимся, что

$$\lim_{\delta \rightarrow 0} |P_\delta - P| = 0. \quad (6.50)$$

Действительно, предположив противное, мы получили бы последовательность $\{P_{\delta_k}\}$, сходящуюся к точке $\bar{P} \neq P$. С другой стороны, поскольку $\lim_{\delta \rightarrow 0} \rho(K_P^{(\delta)}, P_\delta) = 0$ и K_P — замкнутое множество, $\bar{P} \in K_P$. Но по предположению общая часть K и K_P содержит лишь точку P . Полученное противоречие говорит о справедливости соотношения (6.50). Выберем теперь значение $\delta = \delta^*$ настолько малым, чтобы $|P - P_{\delta^*}| < \epsilon$, где ϵ — величина, фигурирующая в лемме 2. Согласно лемме 2 гиперплоскость Π_{δ^*} является опорной для K в точке P . Положим, что уравнение этой гиперплоскости имеет вид $(\Lambda, X) = 0$.

Пусть Q — произвольная точка K_P , не совпадающая с P . Тогда найдется такое $\mu > 0$, при котором

$$Q_\mu = P + \mu(Q - P) \in K_P^{(\delta^*)},$$

т. е.

$$(\Lambda, Q_\mu) > 0.$$

Поскольку

$$(\Lambda, Q_\mu) = (\Lambda, P) + \mu(\Lambda, Q - P) = \mu(\Lambda, Q) \text{ и } \mu > 0,$$

$$(\Lambda, Q) = \frac{1}{\mu}(\Lambda, Q_\mu) > 0.$$

Таким образом, гиперплоскость Π_{δ^*} удовлетворяет всем требованиям леммы и, следовательно, является искомой. Лемма доказана.

§ 2. ДОКАЗАТЕЛЬСТВА ТЕОРЕМ ДВОЙСТВЕННОСТИ

В этом параграфе приводятся доказательства обеих теорем двойственности, сформулированных в п. 1.2 гл. 4. Следует отметить, что все дальнейшие рассуждения существенно опираются на результаты последнего пункта предыдущего параграфа.

Доказательство первой теоремы двойственности. Допустим, что задача I (п. 1.2 гл. 4) разрешима. Через $\bar{A}_j$ обозначим $(m + 1)$ -мерный вектор, первые m компонент которого совпадают с соответствующими компонентами вектора условий задачи I A_j , а последняя равна коэффициенту линейной формы этой задачи c_j , $j = 1, 2, \dots, n$ ($\bar{A}_j$ — расширенный вектор условий задачи I). Положим далее $\bar{B} = (b_1, \dots, b_m, \Delta)$, где $B = (b_1, \dots, b_m)$ — вектор ограничений рассматриваемой задачи, а Δ — максимальное значение ее линейной формы. С помощью оптимального плана задачи I $X^* = (x_1^*, x_2^*, \dots, x_n^*)$ вектор $\bar{B}$ может быть представлен в виде

$$\bar{B} = \sum_{j=1}^n x_j^* \bar{A}_j + \sum_{i=1}^m \varepsilon_i \bar{e}_i. \quad (6.51)$$

Здесь под $\bar{e}_i$ понимается $(m + 1)$ -мерный вектор, i -я компонента которого равна 1, а остальные — нулю;

$$\varepsilon_i = b_i - \sum_{j=1}^n a_{ij} x_j^* \geq 0, \quad i = 1, 2, \dots, m.$$

Пусть K — выпуклый многогранный конус с вершиной в начале координат и образующими векторами $\bar{A}_1, \bar{A}_2, \dots, \bar{A}_n, \bar{e}_1, \bar{e}_2, \dots, \bar{e}_m$. Соотношение (6.51) означает, что $\bar{B}$ принадлежит K . Поскольку X^* — оптимальный план задачи I, при любом $\mu > 0$ вектор $\bar{B} + \mu \bar{e}_{m+1}$ расположен вне K [в противном случае Δ не было бы максимумом линейной формы (4.3) при ограничениях (4.1), (4.2)]. Воспользуемся теперь леммой 3 п. 1.4, положив $P = \bar{B}$, $N_1 = 1$, $B_1 = \bar{e}_{m+1}$. Тогда найдется гиперплоскость Π с уравнением $(\Lambda, Z) = 0$, обладающая свойствами:

- а) $(\Lambda, \bar{A}_j) \leq 0, \quad j = 1, 2, \dots, n;$
- б) $(\Lambda, \bar{e}_i) \leq 0, \quad i = 1, 2, \dots, m;$
- в) $(\Lambda, \bar{B}) = 0;$
- г) $(\Lambda, \bar{B} + \bar{e}_{m+1}) = (\Lambda, \bar{e}_{m+1}) > 0.$

Первые два условия а), б) указывают на опорность Π по отношению к конусу K . Равенство в) означает, что гиперплоскость Π содержит точку B .

Неравенство г) является аналогом условия (6.49) (конус K_P в данном случае совпадает с лучом $\bar{B} + \mu \bar{e}_{m+1}, \mu \geq 0$). В силу г) $(m+1)$ -я компонента вектора $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_m, \lambda_{m+1})$ положительна. Учитывая этот факт определим m -мерный вектор $Y^* = (y_1^*, y_2^*, \dots, y_m^*)$, положив

$$y_i^* = -\frac{\lambda_i}{\lambda_{m+1}}, \quad i = 1, 2, \dots, m.$$

Преобразуя очевидным образом условия а), б), получаем

$$\sum_{i=1}^m a_{ij} y_i^* \geq c_j, \quad j = 1, 2, \dots, n;$$

$$y_i^* \geq 0, \quad i = 1, 2, \dots, m.$$

Таким образом, $Y^* = (y_1^*, y_2^*, \dots, y_m^*)$ — план сопряженной задачи II. Убедимся, что этот план является оптимальным.

Действительно, расписывая в координатах равенство в), имеем

$$\sum_{i=1}^m y_i^* b_i = \Delta,$$

или, поскольку согласно условию $\Delta = \sum_{j=1}^n c_j x_j^*$,

$$\sum_{j=1}^n x_j^* c_j = \sum_{i=1}^m y_i^* b_i. \quad (6.52)$$

Учитывая далее, что для любого плана $Y = (y_1, y_2, \dots, y_m)$ задачи II

$$\sum_{i=1}^m y_i b_i \geq \sum_{j=1}^n x_j^* c_j \text{ (п. 1.1 гл. 4),}$$

приходим к выводу об оптимальности построенного плана Y^* .

Итак, мы определили оптимальный план сопряженной задачи II, для которого имеет место равенство (6.52). Отсюда с очевидностью следует справедливость равенства (6.52) для любой пары оптимальных планов взаимосопряженных задач I, II. Первая теорема двойственности полностью доказана.

Доказательство второй теоремы двойственности. Будем придерживаться обозначений принятых в процессе доказательства предыдущей теоремы.

1. Покажем вначале, что каждому свободному условию задачи I соответствует закрепленное условие задачи II.

Для определенности дадим доказательство для строчного условия задачи (рассмотрение столбцовых условий проводится аналогично)*).

Пусть i_0 -е строчное условие задачи I свободное. По определению это означает существование такого оптимального плана задачи I $X^* = (x_1^*, x_2^*, \dots, x_n^*)$, что

$$\sum_{j=1}^n a_{i_0 j} x_j^* < b_{i_0}.$$

Следовательно, коэффициент

$$\varepsilon_{i_0} = b_{i_0} - \sum_{j=1}^n a_{i_0 j} x_j^*,$$

*) Строчными условиями задач I, II называются неравенства, соответствующие строкам табл. 4.1 (стр. 221): $\sum_{j=1}^n a_{ij} x_j \leq b_i, y_i > 0$.

Неравенства систем (4.2), (4.4), отвечающие столбцам табл. 4.1, именуются столбцовыми условиями двойственной пары I, II.

стоящий при векторе $\bar{e}_{i_0}$ в представлении (6.51), положителен. Рассмотрим произвольный оптимальный план сопряженной задачи II

$$Y^* = (y_1^*, y_2^*, \dots, y_m^*).$$

Если положить

$$\Lambda = (-y_1^*, -y_2^*, \dots, -y_m^*, 1),$$

то, так как Y^* — план задачи II,

$$\begin{aligned} (\Lambda, \bar{A}_j) &\leq 0, \quad j=1, 2, \dots, n; \\ (\Lambda, \bar{e}_i) &\leq 0, \quad i=1, 2, \dots, m. \end{aligned} \quad (6.53)$$

Учитывая далее оптимальность плана Y^* , приходим к равенству

$$\Delta = \sum_{j=1}^n c_j x_j^* = \sum_{i=1}^m b_i y_i^*,$$

или, что то же самое,

$$(\Lambda, \bar{B}) = 0. \quad (6.54)$$

Подставляя в (6.54) выражение для $\bar{B}$, определяемое соотношением (6.51), имеем

$$\sum_{j=1}^n x_j^* (\Lambda, \bar{A}_j) + \sum_{i=1}^m \epsilon_i (\Lambda, \bar{e}_i) = 0. \quad (6.55)$$

В соответствии с неравенствами (6.53) каждое слагаемое левой части (6.55) неположительно. Поэтому для соблюдения равенства (6.55) необходимо, чтобы все они равнялись нулю. В частности,

$$\epsilon_{i_0} (\Lambda, \bar{e}_{i_0}) = 0.$$

Но $\epsilon_{i_0} > 0$. Следовательно,

$$(\Lambda, \bar{e}_{i_0}) = -y_{i_0}^* = 0.$$

Закрепленность i_0 -го условия (строчного) задачи II доказана.

2. Покажем теперь, что каждому закрепленному условию задачи I соответствует свободное условие за-

дачи II. Как и в первой части доказательства, рассмотрение столбцовых и строчных условий проводится аналогично. Пусть, например, столбцовое условие задачи I с номером j_0 является закрепленным. Согласно определению закрепленности условия это означает, что в любом оптимальном плане задачи I j_0 -я компонента равна нулю, т. е. в произвольном представлении вектора $\bar{B}$ вида (6.51) вектор $\bar{A}_{j_0}$ отсутствует. Рассмотрим выпуклый многогранный конус $K_{\bar{B}}$ с вершиной в точке $\bar{B}$, образованный векторами:

$$\bar{e}_{m+1}; (\bar{B} - \bar{A}_{j_0}).$$

Пусть

$$Q = \bar{B} + \mu_1 (\bar{B} - \bar{A}_{j_0}) + \mu_2 \bar{e}_{m+1} \quad (6.56)$$

($\mu_1 \geq 0, \mu_2 \geq 0$) — произвольная точка $K_{\bar{B}}$. Мы утверждаем, что при $\mu_1 + \mu_2 > 0$ $Q \in K$.

Действительно, для $\mu_1 = 0$ это было установлено при доказательстве первой теоремы двойственности. Пусть теперь $\mu_1 > 0$. Допустим противное:

$$Q = \bar{B} + \mu_1 (\bar{B} - \bar{A}_{j_0}) + \mu_2 \bar{e}_{m+1} \notin K,$$

т. е.

$$Q = \sum_{j=1}^n x_j' \bar{A}_j + \sum_{i=1}^m \varepsilon_i' \bar{e}_i, \quad (6.57)$$

где $x_j' \geq 0, j = 1, 2, \dots, n; \varepsilon_i' \geq 0, i = 1, 2, \dots, m$. Образуем вектор

$$R = \frac{\mu_1}{1 + \mu_1} \bar{A}_{j_0} + \frac{1}{1 + \mu_1} Q. \quad (6.58)$$

Поскольку K — выпуклое множество, $R \in K$. Подставляя в (6.58) вместо вектора Q его выражение из (6.56), имеем

$$R = \bar{B} + \frac{\mu_2}{1 + \mu_1} \bar{e}_{m+1} \in K.$$

Если $\mu_2 > 0$, полученное условие противоречит тому, что Δ — максимальное значение линейной формы зада-

чи I (доказательство первой теоремы двойственности). Если же $\mu_2 = 0$,

$$\bar{B} = R = \frac{\mu_1}{1 + \mu_1} \bar{A}_{j_0} + \frac{1}{1 + \mu_1} Q.$$

Откуда, используя соотношение (6.57), получаем представление для $\bar{B}$ вида (6.51) с положительным коэффициентом при $\bar{A}_{j_0}$, что противоречит допущению о закреплённости j_0 -го столбцового условия задачи I. Итак, все точки конуса $K_{\bar{B}}$, кроме вершины $\bar{B}$, лежат вне K . Заметим теперь, что при любых $\alpha \geq 0$, $\beta \geq 0$, $\alpha + \beta > 0$

$$\alpha \bar{e}_{m+1} + \beta (\bar{B} - \bar{A}_{j_0}) \neq 0.$$

Действительно, предполагая противное, имеем

$$\bar{A}_{j_0} = \bar{B} + \frac{\alpha}{\beta} \bar{e}_{m+1}$$

(очевидно, $\beta > 0$). Полученное равенство показывает, что вектор $\tilde{X} = (0, 0, \dots, 0, \underbrace{1}_{j_0}, 0, \dots, 0, 0)$ является

планом задачи I, причем значение линейной формы (4.3) на нем

$$\sum_{j=1}^n c_j \tilde{x}_j = c_{j_0} = \Delta + \frac{\alpha}{\beta}.$$

Учитывая, что Δ — максимальное значение линейной формы (4.3) при условиях (4.1), (4.2), а $\frac{\alpha}{\beta}$ — неотрицательное число, имеем

$$\alpha = 0.$$

Следовательно, $\bar{A}_{j_0} = \bar{B}$, т. е. $\tilde{X}$ является оптимальным планом задачи I, что противоречит закреплённости столбцового условия задачи I с номером j_0 . Проведенные рассуждения дают основание применить лемму 3 п. 1.4, положив

$$P = \bar{B}, \quad N_1 = 2, \quad B_1 = \bar{e}_{m+1}, \quad B_2 = \bar{B} - \bar{A}_{j_0}.$$

Согласно лемме 3 существует гиперплоскость Π $(\Lambda, Z) = 0$, удовлетворяющая условиям:

$$\left. \begin{array}{l} \text{а) } (\Lambda, \bar{A}_j) \leq 0, \quad j = 1, 2, \dots, n; \\ \text{б) } (\Lambda, \bar{e}_i) \leq 0, \quad i = 1, 2, \dots, m; \\ \text{в) } (\Lambda, \bar{B}) = 0; \end{array} \right\} \begin{array}{l} \text{Гиперплоскость } \Pi \\ \text{является опорной} \\ \text{для } K \text{ в точке } \bar{B}. \end{array}$$

$$\left. \begin{array}{l} \text{г) } (\Lambda, \bar{B} + \bar{e}_{m+1}) = (\Lambda, \bar{e}_{m+1}) > 0; \\ \text{д) } (\Lambda, \bar{B} + (\bar{B} - A_{j_0})) = -(\Lambda, A_{j_0}) > 0. \end{array} \right\} \begin{array}{l} \text{Конус } K_{\bar{B}} \\ \text{всеми своими} \\ \text{точками,} \\ \text{кроме } \bar{B}, \text{ рас-} \\ \text{положен} \\ \text{„над“ } \Pi. \end{array}$$

В соответствии с условием г) $(m+1)$ -я компонента вектора $\Lambda = (\lambda_1, \lambda_2, \dots, \lambda_{m+1})$ положительна, что позволяет ввести вектор $Y^* = (y_1^*, y_2^*, \dots, y_m^*)$, положив

$$y_i^* = -\frac{\lambda_i}{\lambda_{m+1}}, \quad i = 1, 2, \dots, m.$$

Используя условия а), б), в), г) подобно тому, как это было сделано при доказательстве первой теоремы двойственности, заключаем, что Y^* — оптимальный план сопряженной задачи II. Согласно условию д)

$$\sum_{i=1}^m a_{ij_0} \lambda_i + c_{j_0} \lambda_{m+1} < 0.$$

Откуда, учитывая положительность λ_{m+1} , приходим к неравенству

$$\sum_{i=1}^m a_{ij_0} y_i^* > c_{j_0}. \quad (6.59)$$

Итак, получен оптимальный план задачи II, удовлетворяющий неравенству (6.59).

Следовательно, j_0 -е столбцовое условие задачи II является свободным. Вторая теорема двойственности доказана.

§ 3. ОБОСНОВАНИЕ НЕКОТОРЫХ УТВЕРЖДЕНИЙ ГЛ. 5

В этом параграфе будет доказан ряд фактов, относящихся к транспортной задаче. Все они сформулированы в соответствующих параграфах гл. 5.

3.1. Свойства планов транспортной задачи (дополнение к § 1 гл. 5).

1. Доказательство критерия опорности плана (теорема 1).

Достаточность. Пусть замкнутых путей, состоящих из коммуникаций с отличными от нуля перевозками (запланированными в соответствии с $X = \|x_{ij}\|_{m,n}$), не существует.

Докажем, что план $X = \|x_{ij}\|_{m,n}$ является опорным. Для этого предположим противное, т. е. линейную зависимость системы векторов P_{ij} , соответствующих положительным перевозкам. Через E обозначим совокупность пар индексов (i, j) , для которых $x_{ij} > 0$. Тогда найдутся такие числа a_{ij} , $i, j \in E$, среди которых имеются не равные нулю, что

$$\sum_{i,j \in E} a_{ij} P_{ij} = 0. \quad (6.60)$$

Пусть $a_{i_1 j_1} \neq 0$ ($i_1, j_1 \in E$). В таком случае из (6.60) получаем

$$-a_{i_1 j_1} P_{i_1 j_1} = \sum_{i,j \in E_1} a_{ij} P_{ij}, \quad (6.61)$$

где E_1 образуется из E вычеркиванием пары индексов (i_1, j_1) . В силу равенства (6.61) среди векторов P_{ij} , $i, j \in E_1$ найдется хотя бы один вектор вида $P_{i_1 j_2}$ с коэффициентом $a_{i_1 j_2} \neq 0$. Перенесем этот вектор в левую часть равенства (6.61)

$$-a_{i_1 j_1} P_{i_1 j_1} - a_{i_1 j_2} P_{i_1 j_2} = \sum_{i,j \in E_2} a_{ij} P_{ij}, \quad (6.62)$$

где $E_2 = E_1 - (i_1, j_2)$.

Поскольку $j_2 \neq j_1$, среди векторов правой части (6.62) обязательно найдется хотя бы один вида $P_{i_2 j_2}$, где $a_{i_2 j_2} \neq 0$. Перенесем его также в левую часть (6.62)

$$-a_{i_1 j_1} P_{i_1 j_1} - a_{i_1 j_2} P_{i_1 j_2} - a_{i_2 j_2} P_{i_2 j_2} = \sum_{i,j \in E_3} a_{ij} P_{ij}, \quad (6.63)$$

где $E_3 = E_2 - (i_2, j_2)$.

Пусть проведено $2k - 1$ шагов (для определенности число шагов выбрано нечетным). Тогда

$$-\sum_{\mu=1}^k a_{i_\mu j_\mu} P_{i_\mu j_\mu} - \sum_{\mu=1}^{k-1} a_{i_\mu j_{\mu+1}} P_{i_\mu j_{\mu+1}} = \sum_{i, j \in E_{2k-1}} a_{ij} P_{ij}, \quad (6.64)$$

где $E_{2k-1} = E_{2k-2} - (i_k, j_k)$.

Возможны два случая:

а) $i_k = i_\lambda$ при некотором λ , $\lambda = 1, 2, \dots, k-1$;

б) $i_k \neq i_\lambda$, $\lambda = 1, 2, \dots, k-1$.

В первом случае а) процесс заканчивается образованием последовательности векторов

$$P_{i_\lambda j_{\lambda+1}}, P_{i_{\lambda+1} j_{\lambda+1}}, \dots, P_{i_{k-1} j_k}, P_{i_\lambda j_k} \quad (i_\lambda = i_k). \quad (6.65)$$

При этом по условию $x_{i_\lambda j_{\lambda+1}} > 0$, $x_{i_{\lambda+1} j_{\lambda+1}} > 0, \dots, \dots, x_{i_{k-1} j_k} > 0$, $x_{i_\lambda j_k} > 0$. Следовательно, из коммуникаций с положительными перевозками можно составить замкнутый маршрут: от A_{i_λ} к $B_{j_{\lambda+1}}$, от $B_{j_{\lambda+1}}$ к $A_{i_{\lambda+1}}$ и т. д., от $A_{i_{k-1}}$ к B_{j_k} , от B_{j_k} к A_{i_λ} .

Во втором случае б) процесс перенесения векторов продолжается. Поскольку $i_k \neq i_\lambda$, $\lambda = 1, 2, \dots, k-1$, среди векторов P_{ij} , $i, j \in E_{2k-1}$ обязательно найдется вектор вида $P_{i_k j_{k+1}}$. Действительно, если бы это было не так, все векторы, участвующие в равенстве (6.64), кроме $P_{i_k j_k}$, имели бы нулевую i_k -ю компоненту, что противоречит (6.64). Описанный процесс не может длиться бесконечно, так как все векторы, переносимые в левую часть, различны. Поэтому через конечное число шагов мы обязательно столкнемся со случаем а), который ведет к построению замкнутого пути из коммуникации с положительными величинами перевозок.

Итак, допустив неопорность плана $X = \|x_{ij}\|_{m,n}$, мы входим в противоречие с условием теоремы. Достаточность критерия доказана.

Необходимость. Пусть $X = \|x_{ij}\|_{m,n}$ — опорный план задачи T . Если бы существовала замкнутая система коммуникаций $A_{i_1} \rightarrow B_{j_1}; B_{j_1} \rightarrow A_{i_2}; \dots; A_{i_s} \rightarrow B_{j_s}; B_{j_s} \rightarrow A_{i_1}$, причем $x_{i_1 j_1} > 0; x_{i_2 j_2} > 0; \dots; x_{i_s j_s} > 0$;

$x_{i_1 j_s} > 0$, то, поскольку $P_{i_1 j_1} - P_{i_2 j_1} + \dots + P_{i_s j_s} - P_{i_1 j_s} = 0$, план X не был бы опорным. Полученное противоречие доказывает вторую часть теоремы.

2. Доказательство теоремы о структуре опорных планов задачи T (теорема 3). Установим прежде всего одно полезное свойство матрицы A , составленной из векторов коммуникаций P_{ij} задачи T .

Лемма. *Любой минор матрицы A равен либо 0, либо ± 1 .*

Доказательство проведем методом математической индукции. Для миноров A первого порядка утверждение леммы тривиально, поскольку элементы матрицы A — нули и единицы. Предположим, что лемма верна для всех миноров A порядка $k-1$, и докажем ее справедливость для любого минора порядка k . Разделим строки матрицы A на две группы: первые m строк отнесем к 1-й группе, последующие n строк — ко 2-й. Из вида матрицы A следует, что каждый ее столбец содержит одну единицу среди строк 1-й группы, другую — среди строк 2-й группы. Пусть Δ_k — произвольный минор матрицы A порядка k . Каждый столбец Δ_k может содержать либо две единицы, либо одну единицу, либо сплошь состоять из нулей. Если имеет место последняя из перечисленных возможностей, утверждение леммы очевидно, так как $\Delta_k = 0$. Пусть теперь каждый столбец Δ_k содержит по меньшей мере одну единицу. Тогда могут представиться два случая:

а) хотя бы в одном столбце Δ_k содержится ровно одна 1;

б) во всех столбцах Δ_k — по две единицы.

Разберем оба случая.

а) Выбираем столбец Δ_k , содержащий одну единицу, и раскладываем по нему исследуемый минор. В результате получаем

$$\Delta_k = \pm \Delta'_{k-1},$$

где Δ'_{k-1} — некоторый минор матрицы A порядка $k-1$. Откуда по предположению индукции следует, что Δ_k равен либо 0, либо ± 1 .

б) В этом случае среди строк Δ_k имеются представители как первой, так и второй групп строк матрицы A . Выбираем любую строку Δ_k , принадлежащую к первой

группе, и прибавляем к ней остальные строки этой группы. В результате образуется строка, состоящая из одних единиц (каждый столбец Δ_k на пересечении со строками 1-й группы содержит ровно одну единицу). Аналогично, прибавляя к одной из строк второй группы остальные строки этой группы, получаем на ее месте еще одну единичную строку. Заметим, что проведенные преобразования не меняют величины Δ_k . Итак, Δ_k совпадает с определителем, обладающим двумя одинаковыми строками, и, следовательно, равен нулю. Таким образом, лемма доказана полностью.

Переходим к доказательству теоремы. Допустим, что исследуемый опорный план $X = \|x_{ij}\|$ состоит из l положительных перевозок. Поскольку векторы P_{ij} , соответствующие этим перевозкам, линейно независимы (данный план — опорный!), из матрицы, составленной ими, можно выделить квадратную подматрицу A_l порядка l с определителем, отличным от нуля. Таким образом, l положительных перевозок x_{ij} рассматриваемого плана удовлетворяют системе l линейных уравнений с матрицей A_l и правыми частями, являющимися подсистемой величин: $a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n$. Положительные перевозки опорного плана X определяются из полученной системы по правилу Крамера. Перенумеруем положительные элементы $X = \|x_{ij}\|_{m,n}$ согласно их месту в рассматриваемой системе. Тогда j -й элемент равняется дроби, числителем которой является определитель матрицы, полученной из A_l заменой ее j -го столбца столбцом свободных членов, а знаменателем — определитель матрицы $A_l: |A_l|$. В соответствии с леммой $|A_l| = \pm 1$. Раскладывая далее определитель, стоящий в числителе, по элементам j -го столбца и учитывая, что миноры матрицы A_l порядка $l-1$ являются минорами A и, следовательно, равны либо 0, либо ± 1 , получаем утверждение теоремы для произвольной положительной перевозки. Теорема доказана.

3.2. К обоснованию метода потенциалов (дополнение к § 2 гл. 5). Установим прежде всего связь между оптимальностью и потенциальностью произвольного плана транспортной задачи T .

Теорема (критерий оптимальности). *Потенциальность плана транспортной задачи T является необходимым и достаточным условием его оптимальности.*

Достаточность. Пусть $X = \|x_{ij}\|_{m,n}$ потенциальный план задачи T . По определению это означает существование таких величин:

$$u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n,$$

что

$$\begin{aligned} \text{а) } v_j - u_i &\leq c_{ij}, \quad i = 1, 2, \dots, m, \\ j &= 1, 2, \dots, n; \end{aligned} \quad (6.66)$$

$$\text{б) } v_j - u_i = c_{ij} \text{ для } (i, j) \in E_X. \quad (6.67)$$

Здесь через E_X обозначено множество пар индексов (i, j) , таких, что $x_{ij} > 0$.

Оценим суммарные транспортные издержки произвольного плана $X' = \|x'_{ij}\|_{m,n}$. Используя неравенство (6.66), получаем

$$\begin{aligned} L_{X'} &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x'_{ij} \geq \sum_{i=1}^m \sum_{j=1}^n (v_j - u_i) x'_{ij} = \\ &= \sum_{i=1}^n v_j \sum_{i=1}^m x'_{ij} - \sum_{i=1}^m u_i \sum_{j=1}^n x'_{ij}. \end{aligned}$$

Поскольку X' — план задачи T ,

$$\sum_{i=1}^m x'_{ij} = b_j, \quad j = 1, 2, \dots, n;$$

$$\sum_{j=1}^n x'_{ij} = a_i, \quad i = 1, 2, \dots, m.$$

Следовательно,

$$L_{X'} \geq \sum_{j=1}^n v_j b_j - \sum_{i=1}^m u_i a_i. \quad (6.68)$$

Принимая теперь во внимание условия (6.67), имеем

$$\begin{aligned} L_X &= \sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = \sum_{i=1}^m \sum_{j=1}^n (v_j - u_i) x_{ij} = \\ &= \sum_{j=1}^n v_j b_j - \sum_{i=1}^m u_i a_i. \end{aligned} \quad (6.69)$$

Сравнение соотношений (6.68) и (6.69) приводит к неравенству

$$L_{X'} \geq L_X,$$

справедливому для любого плана X' исследуемой задачи. Поэтому $X = \|x_{ij}\|_{m,n}$ — оптимальный план задачи T .

Необходимость. Рассмотрим задачу линейного программирования, сопряженную исследуемой задаче T . Согласно общим правилам, изложенным в § 1 гл. 4, она может быть сформулирована так.

Обратить в максимум линейную форму

$$\sum_{j=1}^n b_j v_j - \sum_{i=1}^m a_i u_i \quad (6.70)$$

при условиях

$$v_j - u_i \leq c_{ij}, \quad i = 1, 2, \dots, m, \quad j = 1, 2, \dots, n. \quad (6.71)$$

Обозначим эту задачу через T^* .

В силу первой теоремы двойственности из разрешимости задачи T следует разрешимость задачи T^* . Пусть вектор $W = (\bar{u}_1, \bar{u}_2, \dots, \bar{u}_m, \bar{v}_1, \bar{v}_2, \dots, \bar{v}_n)$ является решением задачи T^* . Докажем теперь потенциальность любого оптимального плана задачи T . Через E обозначим совокупность пар индексов (i, j) , определяемую условием:

$(i, j) \in E$ в том и только в том случае, если существует такой оптимальный план $X = \|x_{ij}\|_{m,n}$ задачи T , что $x_{ij} > 0$. По второй теореме двойственности

$$\bar{v}_j - \bar{u}_i = c_{ij} \text{ для } (i, j) \in E. \quad (6.72)$$

Пусть $X = \|x_{ij}\|_{m,n}$ — произвольное решение задачи T . Так как определенное ранее множество пар индексов $E_X \subset E$, равенство (6.72) имеет место для всех $(i, j) \in E_X$, что в совокупности с системой неравенств (6.71), которым удовлетворяют величины $\bar{u}_1, \bar{u}_2, \dots, \bar{u}_m, \bar{v}_1, \bar{v}_2, \dots, \bar{v}_n$, означает потенциальность плана X . Теорема полностью доказана.

Установленная теорема, в частности, показывает, что потенциал задачи T $W = (u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n)$ может быть связан (соотношением (6.67)) лишь с оптимальным ее планом. Будем именовать этот

план соответствующим данному потенциалу. Пусть W — потенциал задачи T , которому соответствует оптимальный план $X = \|x_{ij}\|_{m,n}$. Обозначим через S совокупность всех оптимальных планов задачи T и покажем, что любой из них соответствует потенциалу W . Так как $u_i, i = 1, 2, \dots, m, v_j, j = 1, 2, \dots, n$, удовлетворяют неравенствам (6.66), вектор $W = (u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n)$ является планом задачи T^* . Далее, поскольку план $X = \|x_{ij}\|_{m,n}$ соответствует потенциалу W ,

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij} = \sum_{j=1}^n b_j v_j - \sum_{i=1}^m a_i u_i. \quad (6.73)$$

Следовательно, $W = (u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n)$ — решение задачи T^* (следствие первой теоремы двойственности). Пусть теперь $\bar{X} = \|\bar{x}_{ij}\|_{m,n}$ — произвольный оптимальный план задачи T . Под E и $E_{\bar{X}}$ будем понимать множества пар индексов (i, j) , определенные в процессе доказательства критерия оптимальности. Согласно второй теореме двойственности

$$v_j - u_i = c_{ij} \text{ для } (i, j) \in E. \quad (6.74)$$

Но так как X — оптимальный план, $E_{\bar{X}} \subset E$, и равенства (6.74) справедливы, в частности, для всех $(i, j) \in E_{\bar{X}}$. Следовательно, план $\bar{X}$ соответствует потенциалу W . Таким образом, потенциал транспортной задачи определяется не отдельным ее оптимальным планом, а всей их совокупностью.

Переходим к выяснению связи между методом потенциалов и методом улучшения плана (гл. 3). Попутно будут обоснованы отдельные элементы метода потенциалов, доказательства которых в гл. 5 были опущены.

Допустим, что $X = \|x_{ij}\|_{m,n}$ — некоторый опорный план задачи T . Через K_X обозначим совокупность векторов коммуникаций P_{ij} , для которых $x_{ij} > 0$. В терминологии общей задачи линейного программирования K_X — базис плана X , векторы P_{ij} — векторы условий. Наши рассуждения будут вестись в предположении невырожденности задачи T . Поэтому K_X состоит из $m + n - 1$

линейно независимых векторов, через которые однозначно выражается любой из векторов P_{ij} . Перенумеруем векторы базиса K_X и через $A_l = (a_{1l}, a_{2l}, \dots, a_{m+n,l})$ обозначим l -й вектор этой совокупности (очевидно, все координаты вектора A_l , кроме двух, равных единице, совпадают с нулем). На первом этапе метода потенциалов определяются величины $u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n$, удовлетворяющие системе уравнений

$$v_j - u_i = c_{ij} \text{ для } (i, j) \in E_X. \quad (6.75)$$

Используя новые обозначения векторов $P_{ij} \in K_X$, систему уравнений (6.75) можно переписать в следующей эквивалентной форме:

$$(A_l, M) = \sum_{i=1}^{n+m} a_{li} \mu_i = c_l, \quad l = 1, 2, \dots, n + m - 1.$$

Здесь $M = (\mu_1, \mu_2, \dots, \mu_{n+m}) = (-u_1, -u_2, \dots, -u_m, v_1, v_2, \dots, v_n)$; c_l — стоимость единичной перевозки в направлении, соответствующем вектору A_l .

Итак, величины $-u_1, -u_2, \dots, -u_m, v_1, v_2, \dots, v_n$ составляют вектор $(\mu_1, \mu_2, \dots, \mu_{n+m})$, определяемый во II алгоритме метода улучшения плана (§ 6 гл. 3). Если для любого вектора P_{ij}

$$(P_{ij}, M) = v_j - u_i \leq c_{ij},$$

план X в соответствии с результатами § 6 гл. 3 является оптимальным. Это же заключение можно сделать из первой части (достаточность) доказанного здесь критерия оптимальности. Пусть при некоторых i, j

$$v_j - u_i > c_{ij}.$$

Тогда в соответствии с методом последовательного улучшения плана выбирается такая пара индексов i_0, j_0 , на которой выражение $c_{ij} - v_j + u_i$ достигает своего минимума, и осуществляется переход к новому базису путем замены одного из векторов системы K_X вектором $P_{i_0 j_0}$. Та же операция проводится и в методе потенциалов (2-й этап). Переход к новому базису связан с разложением вводимого вектора $(P_{i_0 j_0})$ по векторам

старого базиса (K_X), что эквивалентно решению соответствующей системы линейных уравнений. Благодаря простой структуре векторов P_{ij} решение этой системы существенно облегчается по сравнению с общим случаем. Дело сводится к отысканию цепочки из векторов K_X вида

$$P_{i_0j_1}, P_{i_1j_2}, \dots, P_{i_{s-1}j_s}, P_{i_sj_0}. \quad (6.76)$$

Тогда, как легко видеть,

$$P_{i_0j_0} = P_{i_0j_1} - P_{i_1j_1} + \dots - P_{i_{s-1}j_{s-1}} + P_{i_sj_0}. \quad (6.77)$$

Что касается существования указанной цепочки, то оно следует из невырожденности задачи T . В самом деле, если бы в базисе K_X не содержалось ни одной цепочки вида (6.76), то пункты A_{i_0} , B_{j_0} , не были бы связаны системой коммуникаций с ненулевыми согласно плану X перевозками. Поэтому группа пунктов производства, связанных (коммуникациями с положительными перевозками) с A_{i_0} , обслуживала бы не все пункты потребления. Но тогда объем производства в этих пунктах (суммарный) совпадал бы с потребностями части пунктов потребления (без B_{j_0}), что невозможно в силу невырожденности задачи T (п. 1.5 гл. 5). Построение цепочки (6.76) осуществляется на 2-м этапе итерации метода потенциалов. Согласно методу улучшения плана после разложения вводимого вектора (A_j) по векторам старого базиса разыскивается величина

$$\theta = \min_{z_{ij} > 0} \frac{x_i}{z_{ij}}, \quad (6.78)$$

где $x_i(z_{ij})$ — коэффициент при A_i в разложении вектора ограничений (вводимого вектора A_j) по векторам старого базиса; минимум берется по тем i , для которых $z_{ij} > 0$. Если $\theta = \frac{x_r}{z_{rj}}$, вектор A_r из базиса удаляется.

Положительные компоненты нового плана определяются по формулам

$$\left. \begin{aligned} x_i' &= x_i - \theta z_{ij}, & i &= 1, 2, \dots, n + m - 1, & i \neq j; \\ x_j' &= \theta. \end{aligned} \right\} \quad (6.79)$$

В случае транспортной задачи, для которой, как следует из соотношения (6.77), $z_{ij} = 0, \pm 1$, формулы (6.78), (6.79) переходят в формулы (5.20), (5.21), выведенные в § 2 гл. 5. С их помощью на 2-м этапе итерации метода потенциалов строится новый план перевозок. Итак, метод потенциалов включает те же основные операции, что и II алгоритм метода последовательного улучшения плана.

Однако применительно к транспортной задаче большинство из них существенно упрощается. Для завершения обоснования метода потенциалов нам осталось показать, что:

а) при опорности первоначального плана любой последующий план будет также опорным;

б) решение транспортной задачи укладывается в конечное число итераций.

Доказательства этих утверждений можно было бы провести непосредственно. Однако в этом нет необходимости, поскольку при описании метода улучшения плана, идентичность которого с методом потенциалов нами установлена, они были доказаны для общей задачи линейного программирования.

При рассмотрении случая вырожденности транспортной задачи (п. 2.5 гл. 5) мы оставили без доказательства следующее утверждение.

Существует такое $\Delta > 0$, что

а) если $X(\epsilon) = \|x_{ij}(\epsilon)\|_{m,n} = \|x_{ij}\| + \epsilon \|a_{ij}\|$ — опорный план задачи $T(\epsilon)$ и $0 < \epsilon < \Delta$, то при любом $0 \leq \epsilon' < \Delta$ $X(\epsilon') = \|x_{ij}\| + \epsilon' \|a_{ij}\|$ — опорный план задачи $T(\epsilon')$;

б) если $X(\epsilon)$ — решение задачи $T(\epsilon)$, то $X(\epsilon')$ — решение задачи $T(\epsilon')$.

В частности, утверждение справедливо для $\epsilon' = 0$.

Доказательство. а) Согласно определению разложения (5.24)

$$\begin{aligned} \sum_{i=1}^m x_{ij} &= b_j, \quad j = 1, 2, \dots, n; \\ \sum_{j=1}^n x_{ij} &= a_i, \quad i = 1, 2, \dots, m; \\ \sum_{i=1}^m a_{ij} &= \begin{cases} 0, & j \neq n, \\ m, & j = n; \end{cases} \\ \sum_{j=1}^n a_{ij} &= 1, \quad i = 1, 2, \dots, m. \end{aligned}$$

Поэтому при любом ε'

$$\left. \begin{aligned} \sum_{i=1}^m x_{ij}(\varepsilon') &= \begin{cases} b_j, & j=1, 2, \dots, n-1, \\ b_j + m\varepsilon', & j=n \end{cases} \\ \sum_{j=1}^n x_{ij}(\varepsilon') &= a_i + \varepsilon', \quad i=1, 2, \dots, m. \end{aligned} \right\} \quad (6.80)$$

Вспомним, что матрица $\|x_{ij}\|_{m,n}$ образуется в результате разложения по базису плана $X(\varepsilon)$ вектора $P = (a_1, a_2, \dots, a_m, b_1, b_2, \dots, b_n)$. Аналогично, исходя из вектора $Q = (1, 1, \dots, 1, 0, 0, \dots, m)$, составляется матрица $\|\alpha_{ij}\|_{m,n}$. Пусть s — произвольная совокупность $n + m - 1$ линейно независимых векторов коммуникаций задачи T . Через x_s обозначим абсолютную величину минимального по модулю ненулевого коэффициента в разложении вектора P по системе s . Через α_s обозначим абсолютную величину максимального по модулю коэффициента в разложении Q по векторам той же системы. Положим $\Delta_1 = \min_s x_s$, $\Delta_2 = \max_s \alpha_s$, где максимум и минимум берутся по всем линейно независимым системам s . Поскольку число различных систем s конечно, $\Delta_1 > 0$; $\Delta_2 < \infty$.

Пусть $\Delta = \frac{\Delta_1}{\Delta_2}$; $\varepsilon < \Delta$. Рассмотрим произвольный опорный план задачи $T(\varepsilon)$ $X(\varepsilon) = \|x_{ij}\|_{m,n} + \varepsilon \|\alpha_{ij}\|_{m,n}$.

Покажем прежде всего, что $x_{ij} \geq 0$, $i=1, 2, \dots, m$; $j=1, 2, \dots, n$. Действительно, если при некоторых i, j , $x_{ij} < 0$, то согласно определению Δ_1 $x_{ij} \leq -\Delta_1$. Следовательно, $x_{ij}(\varepsilon) \leq -\Delta_1 + \varepsilon\Delta_2 < -\Delta_1 + \frac{\Delta_1}{\Delta_2} \Delta_2 < 0$. Полученное противоречие указывает на неотрицательность всех элементов матрицы $\|x_{ij}\|_{m,n}$.

Пусть теперь $0 \leq \varepsilon' < \Delta$. Тогда $x_{ij}(\varepsilon') = x_{ij} + \varepsilon'\alpha_{ij} \geq \Delta_1 - \varepsilon'\Delta_2 \geq 0$ ($x_{ij} > 0$). При $x_{ij} = 0$

$$x_{ij}(\varepsilon') = \varepsilon'\alpha_{ij} = \frac{\varepsilon'}{\varepsilon} x_{ij}(\varepsilon) \geq 0.$$

Итак, все элементы матрицы $\|x_{ij}(\varepsilon')\|_{m,n}$ неотрицательны и удовлетворяют системе соотношений (6.80). Следовательно, $X(\varepsilon')$ — план задачи $T(\varepsilon')$. Поскольку

базис плана $X(\varepsilon)$ является вместе с тем базисом $X(\varepsilon')$, план $X(\varepsilon')$ опорный.

б) Пусть $X(\varepsilon) \rightarrow$ опорное решение задачи $T(\varepsilon)$, $0 < \varepsilon < \Delta$. При $0 \leq \varepsilon' < \Delta$ $X(\varepsilon') = \|x_{ij}\|_{m,n} + \varepsilon' \|a_{ij}\|_{m,n}$ является опорным планом задачи $T(\varepsilon')$ (а). Покажем, что план $X(\varepsilon')$ оптимальный для задачи $T(\varepsilon')$. $X(\varepsilon)$, будучи оптимальным планом $T(\varepsilon)$, является в силу критерия оптимальности потенциальным планом этой задачи. Поэтому существуют величины $u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n$, такие, что

$$v_j - u_i \leq c_{ij}, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n;$$

$$v_j - u_i = c_{ij}, \quad \text{если } x_{ij}(\varepsilon) > 0. \quad (6.81)$$

Допустим, что $x_{ij}(\varepsilon') > 0$. Тогда рассуждения, аналогичные проведенным в а), показывают, что $x_{ij}(\varepsilon) > 0$. Значит равенства (6.81) справедливы для всех тех i, j , при которых $x_{ij}(\varepsilon') > 0$. Таким образом, система величин $u_1, u_2, \dots, u_m, v_1, v_2, \dots, v_n$ является потенциалом задачи $T(\varepsilon')$, связанным с планом $X(\varepsilon')$. Следовательно, $X(\varepsilon')$ — решение задачи $T(\varepsilon')$.

3.3. Венгерский метод как уточнение метода сокращения невязок применительно к транспортной проблеме (дополнение к § 3 гл. 5). При рассмотрении венгерского метода мы указывали на идейную общность, объединяющую его с методом сокращения невязок (§ 3 гл. 4). Далее устанавливается, что венгерский метод является по существу детализацией метода последовательного сокращения невязок для одной из частных задач линейного программирования — транспортной.

Перепишем транспортную задачу T в векторной форме:

обратить в минимум

$$\sum_{i=1}^m \sum_{j=1}^n c_{ij} x_{ij}$$

при соблюдении условий

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij} P_{ij} = P;$$

$$x_{ij} \geq 0, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n,$$

Здесь, как обычно, P_{ij} — вектор коммуникаций, P — вектор производства — потребления.

Задача T^* , сопряженная T , состоит, как мы видели, в выборе вектора $W = (-u_1, -u_2, \dots, -u_m, v_1, v_2, \dots, v_n)$, максимизирующего линейную форму

$$(W, P) = \sum_{j=1}^n b_j v_j - \sum_{i=1}^m a_i u_i$$

при условиях

$$(W, P_{ij}) \leq c_{ij}, \quad i=1, 2, \dots, m; \quad j=1, 2, \dots, n,$$

которые в силу специфики векторов P_{ij} могут быть переписаны еще и так

$$v_j - u_i \leq c_{ij}, \quad i=1, 2, \dots, m, \quad j=1, 2, \dots, n.$$

Напомним общую схему метода сокращения невязок применительно к задаче T . Вначале разыскивается некоторый план задачи T^* $W_0 = (-u_1^{(0)}, -u_2^{(0)}, \dots, -u_m^{(0)}, v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)})$ и выделяются те из векторов P_{ij} , для которых

$$(W_0, P_{ij}) = v_j - u_i = c_{ij}. \quad (6.82)$$

С помощью выделенных векторов формулируется вспомогательная задача T_{W_0} , связанная с планом W_0 задачи T^* . Она состоит в минимизации линейной формы $\sum_{i=1}^{n+m} \varepsilon_i$ при условиях

$$\sum_{(i,j) \in E_0} x_{ij} P_{ij} + \sum_{l=1}^{n+m} \varepsilon_l e_l = P,$$

$$x_{ij} \geq 0, \quad (i, j) \in E_0; \quad \varepsilon_i \geq 0, \quad i=1, 2, \dots, n+m.$$

Здесь через E_0 обозначена совокупность всех тех пар индексов (i, j) , для которых справедливо равенство (6.82); векторы $e_1, e_2, \dots, e_{n+m}$ составляют полный набор единичных векторов $(n+m)$ -мерного пространства. Поскольку

$$\sum_{l=1}^{n+m} \varepsilon_l = \sum_{i=1}^m a_i + \sum_{j=1}^n b_j - 2 \sum_{(i,j) \in E_0} x_{ij},$$

то сформулированная задача эквивалентна следующей:
максимизировать

$$\sum_{(i,j) \in E_0} x_{ij}$$

при соблюдении условий

$$\sum_{(i,j) \in E_0} x_{ij} P_{ij} \leq P;$$

$$x_{ij} \geq 0, \quad (i, j) \in E_0.$$

Пусть $\|x_{ij}^{(1)}\|_{m,n}$ — решение задачи T_{W_0} (при $(i, j) \in E_0$ $x_{ij}^{(1)}$ полагается равным нулю).

Если

$$\Delta_1 = \sum_{i=1}^m a_i + \sum_{j=1}^n b_j - 2 \sum_{i=1}^m \sum_{j=1}^n x_{ij}^{(1)} = 0,$$

то $X_1 = \|x_{ij}^{(1)}\|_{m,n}$ — искомое решение задачи T .

Если же $\Delta_1 > 0$, с помощью матрицы X_1 строится новый план задачи T^* $W_1 = (-u_1^{(1)}, -u_2^{(1)}, \dots, -u_m^{(1)}, v_1^{(1)}, v_2^{(1)}, \dots, v_n^{(1)})$, увеличивающий ее линейную форму. Затем весь процесс повторяется, отправляясь от найденного вектора W_1 . Через конечное число итераций мы приходим к такому плану W_r задачи T^* , что решение вспомогательной задачи T_{W_r} является планом задачи T ($\Delta_{r+1} = 0$). Полученная матрица совпадает с оптимальным планом исследуемой задачи T .

Покажем, что алгоритм венгерского метода полностью укладывается в изложенную схему. Обозначим через $-u_1^{(0)}, -u_2^{(0)}, \dots, -u_m^{(0)}$ ($v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)}$) величины, которые в предварительном этапе вычитаются из соответствующих строк (столбцов) матрицы C . Очевидно, вектор $W_0 = (-u_1^{(0)}, -u_2^{(0)}, \dots, -u_m^{(0)}, v_1^{(0)}, v_2^{(0)}, \dots, v_n^{(0)})$ является планом задачи T^* (согласно алгоритму $c_{ij}^{(0)} = c_{ij} - (v_j - u_i) \geq 0$).

Таким образом, в предварительном этапе выбирается некоторый план задачи T^* W_0 и выделяются те позиции i, j , для которых

$$c_{ij} - (v_j - u_i) = c_{ij}^{(0)} = 0.$$

Кроме того, в этом же этапе строится исходный план $X_0 = \|x_{ij}^{(0)}\|_{m,n}$ вспомогательной задачи T_{W_0} , определяемой планом W_0 . Пусть $X_1 = \|x_{ij}^{(1)}\|_{m,n}$ — матрица, образуемая из X_0 до первого перехода к этапу 3. Эта матрица строится, вообще говоря, в результате применения некоторого числа этапов 1, 2. В частности, если за первым же этапом 1 следует этап 3, $X_1 = X_0$. Покажем, что матрица X_1 является решением вспомогательной задачи T_{W_0} . Поскольку последний из этапов 1 завершился случаем IV, все нули матрицы C_0 лежат в выделенных линиях. Пусть $i_1, i_2, \dots, i_\lambda; j_1, j_2, \dots, j_\mu$ — индексы выделенных строк и столбцов матрицы C_0 соответственно. Выделим те же линии (строки или столбцы) в матрице X_1 . Согласно правилам выделения (этап 1) $x_{ij}^{(1)} > 0$ лежит в одной из выделенных линий: либо в строке, либо в столбце. Учитывая далее справедливость для выделенных линий равенств

$$\sum_{j=1}^n x_{i_s j}^{(1)} = a_{i_s}, \quad s = 1, 2, \dots, \lambda;$$

$$\sum_{i=1}^m x_{i j_s}^{(1)} = b_{j_s}, \quad s = 1, 2, \dots, \mu,$$

получаем

$$\sum_{i=1}^m \sum_{j=1}^n x_{ij}^{(1)} = \sum_{s=1}^{\lambda} a_{i_s} + \sum_{s=1}^{\mu} b_{j_s}. \quad (6.83)$$

Пусть $\bar{X}_1 = \|\bar{x}_{ij}^{(1)}\|_{m,n}$ — произвольный план задачи T_{W_0} . Поскольку $\bar{x}_{ij}^{(1)} > 0$ лишь при условии $c_{ij}^{(0)} = 0$, все положительные элементы $\bar{X}_1$ расположены в выделенных линиях. Поэтому

$$\sum_{i=1}^m \sum_{j=1}^n \bar{x}_{ij}^{(1)} \leq \sum_{s=1}^{\lambda} \sum_{j=1}^n \bar{x}_{i_s j}^{(1)} + \sum_{s=1}^{\mu} \sum_{i=1}^m \bar{x}_{i j_s}^{(1)} \leq \sum_{s=1}^{\lambda} a_{i_s} + \sum_{s=1}^{\mu} b_{j_s}. \quad (6.84)$$

Сравнение неравенства (6.84) с равенством (6.83) доказывает наше утверждение. Если матрица X_1 — план задачи T ($\Delta_1 = 0$), то она является решением этой задачи. Если же

$$2 \sum_{i=1}^m \sum_{j=1}^n x_{ij}^{(1)} < \sum_{i=1}^m a_i + \sum_{j=1}^n b_j,$$

то необходим переход к этапу 3, в котором разыскивается новый план задачи T^* и, следовательно, ставится новая вспомогательная задача. Рассмотрим задачу $T_{W_0}^*$, сопряженную вспомогательной задаче T_{W_0} , которая состоит в отыскании вектора $\gamma = (\alpha_1, \alpha_2, \dots, \alpha_m; \beta_1, \beta_2, \dots, \beta_n)$, обращающего в максимум форму

$$\sum_{i=1}^m a_i \alpha_i + \sum_{j=1}^n b_j \beta_j$$

при условиях

$$\alpha_i + \beta_j \leq 0 \text{ для } (i, j) \in E_0;$$

$$\alpha_i \leq 1, \quad \beta_j \leq 1, \quad i = 1, 2, \dots, m; \quad j = 1, 2, \dots, n.$$

После того, как найдено решение задачи T_{W_0} , оптимальный план сопряженной задачи $T_{W_0}^*$ определяется автоматически:

$$\alpha_i^{(1)} = \begin{cases} 1, & i \neq i_s, \quad s = 1, 2, \dots, \lambda; \\ -1, & i = i_s, \quad s = 1, 2, \dots, \lambda; \end{cases}$$

$$\beta_j^{(1)} = \begin{cases} 1, & j \neq j_s, \quad s = 1, 2, \dots, \mu; \\ -1, & j = j_s, \quad s = 1, 2, \dots, \mu. \end{cases}$$

В самом деле, позиции $(i, j) \in E_0$ расположены в выделенных линиях. Следовательно, при $(i, j) \in E_0$ величина $\alpha_i + \beta_j$ совпадает либо с 0, либо с -2 , и вектор $\gamma_1 = (\alpha_1^{(1)}, \alpha_2^{(1)}, \dots, \alpha_m^{(1)}; \beta_1^{(1)}, \beta_2^{(1)}, \dots, \beta_n^{(1)})$ является планом задачи $T_{W_0}^*$.

С другой стороны, $x_{ij}^{(1)} > 0$ расположен лишь на одной из выделенных линий, откуда $\alpha_i^{(1)} + \beta_j^{(1)} = 0$ при $x_{ij}^{(1)} > 0$. Кроме того, неравенства

$$\sum_{i=1}^m x_{ij}^{(1)} < b_j, \tag{6.85}$$

$$\sum_{j=1}^n x_{ij}^{(1)} < a_i \tag{6.86}$$

возможны лишь для невыделенных линий. Поэтому для тех $j(i)$, при которых имеет место неравенство (6.85)

[(6.86)], $\beta_j^{(1)} = 1$ ($\alpha_i^{(1)} = 1$). Используя эти условия, не-
трудно прийти к равенству

$$\sum_{i=1}^{m+n} \epsilon_i^{(1)} = \sum_{j=1}^n b_j \beta_j^{(1)} + \sum_{i=1}^m a_i \alpha_i^{(1)*},$$

указывающему на оптимальность построенного плана
задачи $T_{W_0}^*$. В соответствии с методом сокращения не-
вязок определяется новый план задачи T^*

$$W_1 = W_0 + \theta \gamma_1,$$

где

$$\theta = \min_{\alpha_i + \beta_j > 0} \frac{c_{ij}^{(0)}}{\alpha_i + \beta_j}.$$

Но $\alpha_i + \beta_j > 0$ ($\alpha_i + \beta_j = 2$) лишь для позиций (i, j) ,
расположенных вне выделенных линий. Итак,

$$\theta = \frac{1}{2} \min c_{ij}^{(0)} = \frac{1}{2} h,$$

где минимум берется по всем невыделенным элементам
матрицы $\|c_{ij}^{(0)}\|_{m,n}$. Пусть

$$\begin{aligned} \|c_{ij}^{(1)}\|_{m,n} &= \|c_{ij}^{(0)}\|_{m,n} - \|\alpha_i^{(1)} + \beta_j^{(1)}\| \cdot \theta = \\ &= \|c_{ij}\|_{m,n} - \|v_j^{(0)} - u_i^{(0)} + \theta(\alpha_i^{(1)} + \beta_j^{(1)})\|_{m,n} = \\ &= \|c_{ij}\|_{m,n} - \|v_j^{(1)} - u_i^{(1)}\|_{m,n}. \end{aligned}$$

Нулевые элементы матрицы $\|c_{ij}^{(1)}\|_{m,n}$ определяют сово-
купность векторов условий задачи T_{W_1} , связанной с вновь
построенным планом $W_1 = (-u_1^{(1)}, -u_2^{(1)}, \dots, -u_m^{(1)},$
 $v_1^{(1)}, v_2^{(1)}, \dots, v_n^{(1)})$ сопряженной задачи T^* . В соответ-
ствии с определением матрица $C_1 = \|c_{ij}^{(1)}\|_{m,n}$ строится

*) Мы использовали вспомогательную задачу T_{W_0} в первой фор-
мулировке

$$\begin{aligned} \epsilon_i^{(1)} &= a_i - \sum_{j=1}^n x_{ij}^{(1)}, \quad i = 1, 2, \dots, m, \\ \epsilon_{m+j}^{(1)} &= b_j - \sum_{i=1}^m x_{ij}^{(1)}, \quad j = 1, 2, \dots, n. \end{aligned}$$

по матрице $C_0 = \|c_{ij}^{(0)}\|_{m,n}$ следующим образом: из невыделенных элементов C_0 вычитается $h = 2\theta$, к элементам C_0 , лежащим на пересечении двух выделенных линий, h прибавляется; остальные элементы C_0 изменениям не подвергаются. Следовательно, матрица $C_1 = \|c_{ij}^{(1)}\|_{m,n}$ совпадает с результатом применения этапа 3 к матрице $C_0 = \|c_{ij}^{(0)}\|_{m,n}$.

Итак, этап 3 действительно состоит в построении новой вспомогательной задачи в полном соответствии с методом сокращения невязок. Приведенные рассуждения показывают, что венгерский метод в точности следует общей схеме метода сокращения невязок: между двумя соседними этапами 3 решается очередная вспомогательная задача, в каждом из этапов 3 ставится новая вспомогательная задача. Отличительная особенность венгерского метода состоит лишь в оригинальном способе решения вспомогательной задачи, учитывающем специфику транспортной проблемы.

ГЛАВА 7

ЗАКЛЮЧЕНИЕ

Развитие математики в конце XIX и начале XX века складывалось главным образом под влиянием запросов физики. Именно теоретической физике обязаны своими успехами дифференциальная геометрия, векторный и тензорный анализ, теория вероятностей и математическая статистика, теория дифференциальных и интегральных уравнений и т. д. Основным понятием математических наук этого периода являлась непрерывность. Естествознание и техника не ставили перед математикой в широких масштабах комбинаторные задачи, а состояние вычислительной техники не давало оснований рассчитывать на эффективные методы решения этих задач.

В последние десятилетия жизнь выдвинула на первый план проблемы организации производства, планирования народного хозяйства, автоматизации промышленности и управления военной техникой. Естественной реакцией на это явилось бурное развитие кибернетических наук. Вопросы управления энергией и техникой стали столь же актуальными, как и вопросы получения энергии и создания техники. Стало ясно, что экономический эффект от рациональных методов управления и планирования, применяемых в широких масштабах и на высоком уровне, способен в ряде случаев превысить эффект от существенного увеличения мощности. Потребовались новые математические методы, позволяющие анализировать ритм производства, работу нервной системы живого организма, взаимоотношения между людьми и между коллективами, качество современной военной техники. И хотя по-прежнему центральным по-

нятием современной математики остается непрерывность, все большее внимание приходится уделять дискретным задачам комбинаторного типа.

Математики перестали с пренебрежением относиться к множествам «меры нуль», и получение точного числового результата в обозримый срок стало цениться наравне с доказательствами трудных абстрактных утверждений и выводом изящных формул. Кстати, о формулах. Сложные современные задачи, выдвигаемые жизнью перед математикой, как правило, не позволяют рассчитывать на получение формул, связывающих заданные и искомые величины. В тех случаях, когда качественные оценки не удовлетворяют потребителя, можно скорее надеяться на разработку алгоритмов, позволяющих для каждой конкретной системы исходных данных вычислить результат.

Таким образом, роль физики — поставщика новых направлений и новых задач в математике — перешла в последние годы к кибернетике. Появление и развитие цифровой вычислительной техники форсировало создание новых математических методов, которые, в свою очередь, расширили возможности математических машин. Одним из новых направлений в математике является линейное программирование.

В настоящем руководстве изложены практические задачи, вызвавшие к жизни линейное программирование и определившие пути его развития, и общие методы решения этих задач. Не все понятия линейного программирования можно считать полностью установившимися. Методы этой науки еще не являются законченными. Задачи такого порядка, как планирование народнохозяйственной деятельности в масштабе совнархоза, республики или государства в целом, так же как и современные широкие военные задачи, связаны с вычислением или оценкой огромного количества параметров управления. При существующих общих методах решения задач линейного программирования, не учитывающих специфических особенностей перечисленных задач, решение их является чрезвычайно трудоемкой работой даже при использовании самых быстродействующих современных математических машин. Практические задачи требуют также усовершенствования методов линейного программирования и в ряде других направлений.

Наметились и продолжают вырисовываться связи линейного программирования с другими областями современной математики. Естественно, что все эти новые вопросы бурно развивающейся дисциплины не могли найти отражения в предлагаемой книге, претендующей лишь на систематизацию установившихся понятий и отработанных методов.

Тем не менее, для того чтобы читатель мог составить себе общее представление о путях развития и новых возможных приложениях линейного программирования, мы приводим ниже (§ 1) краткий обзор результатов, не вошедших в программу руководства. Наиболее важный из этих вопросов — связь линейного программирования и теории игр — выделен в отдельный параграф (§ 2). § 3 посвящен перспективным направлениям развития линейного программирования. Это, главным образом, работы последних лет.

§ 1. НЕКОТОРЫЕ СПЕЦИАЛЬНЫЕ ВОПРОСЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

1.1. Выпуклое программирование. Методы линейного программирования могут быть использованы для приближенного (а в отдельных случаях и для точного) решения экстремальных задач при некоторых нелинейных показателях качества, когда переменные ограничены линейными условиями. Такие примеры были рассмотрены в § 9 гл. 2 в задаче об определении рациональной системы допусков и в § 13 той же главы, где рассматривалась проблема целераспределения. В каждой из этих задач требовалось обратить в минимум сумму функций, выпуклых относительно каждой переменной при соблюдении линейных условий.

Напомним идею так называемого выпуклого программирования и приведем некоторые замечания о возможностях усовершенствования и обобщения метода.

Пусть требуется приближенно (с точностью до ε) вычислить минимум суммы

$$R = \sum_{i=1}^n \varphi_i(x_i) \quad (7.1)$$

при условиях

$$\sum_{i=1}^n A_i x_i = B, \quad (7.2)$$

$$x_i \geq 0, \quad (i = 1, 2, \dots, n), \quad (7.3)$$

где $\varphi_i(x_i)$ — выпуклые функции; A_i и B — соответственно векторы условий и вектор ограничений.

В каждой конкретной задаче можно из физических соображений определить границы изменения переменных x_i . (Часто диапазон изменения x_i без труда вычисляется из условий (7.2), (7.3)).

Разобьем интервал изменения x_i на k_i частей так, чтобы ломаная, вписанная в кривую $\varphi_i = \varphi_i(x_i)$, аппроксимировала ее с точностью до $\frac{\varepsilon}{n}$. Уравнения прямых, которым принадлежат звенья ломаной, имеют вид

$$y_i = \lambda_{si} x + \mu_{si}, \\ s = 1, 2, \dots, k_i,$$

где λ_{si} и μ_{si} легко вычисляются через значения $\varphi_i(x_i)$ в точках пересечения кривой и ломаной.

Запишем уравнение ломаной, аппроксимирующей кривую $\varphi_i = \varphi_i(x_i)$ в виде

$$y_i = \tilde{\varphi}_i(x_i).$$

Легко видеть, что вычисление минимума

$$L = \sum_{i=1}^n \tilde{\varphi}_i(x_i) = \sum_{i=1}^n y_i$$

при условиях (7.2) и (7.3) является задачей линейного программирования. Действительно, если $\varphi_i(x_i)$ выпуклая функция x_i , то каждая точка, лежащая выше ломаной, вписанной в кривую $\varphi_i = \varphi_i(x_i)$, лежит выше любой прямой, содержащей звено ломаной. Это значит, что область, расположенная над ломаной, может быть охарактеризована системой линейных неравенств

$$y_i \geq \lambda_{si} x_i + \mu_{si}, \\ s = 1, 2, \dots, k_i.$$

Вычисление минимума L сводится, таким образом, к следующей задаче.

Требуется определить минимум линейной формы

$$L = \sum_{i=1}^n y_i$$

при условиях

$$y_i \geq \lambda_{si} x_i + \mu_{si},$$

$$\sum_{i=1}^n A_i x_i = B,$$

$$x_i \geq 0 \quad (i = 1, 2, \dots, n; \quad s = 1, 2, \dots, k_i)$$

Задача содержит $2n$ переменных $(x_1, x_2, \dots, x_n; y_1, y_2, \dots, y_n)$ и $m + \sum_{i=1}^n k_i$ условий.

Поскольку $\tilde{\varphi}_i(x_i)$ с точностью до $\frac{\varepsilon}{n}$ аппроксимирует $\varphi_i(x_i)$, то решение полученной задачи линейного программирования с точностью до ε соответствует решению исходной нелинейной задачи.

В тех случаях, когда требуется вычислить экстремум R (7.1) на конечном дискретном множестве точек, оптимальный план задачи линейного программирования может определить точное решение нелинейной задачи. С одной из таких задач мы встречались в § 13 гл. 2.

Ясно, что все рассуждения, проведенные здесь относительно минимума суммы выпуклых функций, сохраняют силу для решения задач на максимум линейной комбинации вогнутых функций при линейных ограничениях.

В [27] разработан прием выпуклого программирования, при котором вычисления на каждом шаге сравнимы по трудоемкости с расчетами, необходимыми для решения обычной задачи линейного программирования с n переменными и условиями вида (7.2), (7.3) без дополнительных ограничений. Однако число шагов — итераций при этом увеличивается.

Приведенные здесь построения могут быть обобщены на случай, когда показатели качества, подлежащие оптимизации, представляют собой не сумму выпуклых функций отдельных переменных, а выпуклую функцию относительно совокупности переменных. Но количество

дополнительных условий быстро растет с усложнением показателя качества.

1.2. Целочисленное программирование. Часто физический смысл переменных задачи линейного программирования требует целочисленности искомых параметров управления. Но далеко не во всех случаях можно гарантировать целочисленность решения задачи, не вводя дополнительных условий. Так, проблема выбора (§ 3 гл. 5) решается в целых числах. Но в задаче о распределении самолетов между воздушными линиями (§ 3 гл. 2) или в задаче о размещении оборудования (§ 5 гл. 2) компоненты оптимального плана могут оказаться дробными. В некоторых случаях (если, например, значения x_i достаточно велики, $x_i \gg 1$) можно в качестве приближенного решения принять ближайшие к вычисленным компонентам оптимального плана целые числа. Заметим, что полученное таким образом приближенное решение должно удовлетворять условиям задачи. Нетрудно, однако, указать примеры, в которых округление приводит к решению, весьма далекому от оптимума. Если, например, параметры управления x_i по своему определению могут принимать только два значения 0 и 1, то полученная в результате решения задачи величина $x = \frac{1}{2}$ может с одинаковым успехом быть округлена как в одну, так и в другую сторону. Нетрудно догадаться, что полученное при этом решение может существенно отличаться от оптимального.

Из приведенных рассуждений вытекает актуальность задачи о выборе дополнительных линейных ограничений, обеспечивающих целочисленность решения задачи. Один из методов подбора дополнительных условий [34] рассмотрен ниже.

Пусть в результате решения задачи линейного программирования получен ее оптимальный опорный план

$$X = (x_1, x_2, \dots, x_m, 0, \dots, 0).$$

Пусть при этом оказалось, что, вопреки физическому смыслу искомых переменных, некоторые из них оказались дробными.

Введем дополнительное условие

$$\sum_{i=1}^{n-m} x_{i+m} \geq 1. \quad (7.4)$$

План X не удовлетворяет условию (7.4), так как все его компоненты с индексами, превышающими m , равны нулю. С другой стороны, интересующее нас решение в целых числах удовлетворяет дополнительному условию. Действительно, при целых значениях x_i сумма, записанная в левой части условия (7.4), так же должна быть целым числом. Поскольку $x_i \geq 0$, то любой план, для которого

$$\sum_{i=1}^{n-m} x_{i+m} = 0,$$

должен совпадать с X . Наименьшее значение суммы $\sum_{i=1}^{n-m} x_i$ для целочисленного плана равно единице. Поэтому всякий целочисленный план удовлетворяет неравенству (7.4).

Найдем теперь новое решение задачи линейного программирования, учитывающее условие (7.4) *. Если и оно содержит дробные компоненты, используем снова ту же идею, вводя еще одно условие, подобное ограничению (7.4). Доказано, что таким образом можно в конечное число шагов обеспечить целочисленность решения.

В работах [57] и [47] изложены другие приемы составления дополнительных условий. Меняя вид дополнительных ограничений, можно в конкретных задачах ускорить приближение к целочисленному решению. К сожалению, до сих пор не существует достаточно общих рекомендаций по рациональному подбору дополнительных условий.

1.3. Учет специфики задачи. При большом количестве переменных и условий алгоритмы всех современных общих методов линейного программирования приводят к весьма трудоемким вычислениям. Используя особенности конкретных задач, как это, например, сделано при решении транспортной задачи и проблемы выбора, можно существенно упростить расчеты. Ограничения

* При решении задачи с дополнительным условием естественно воспользоваться одним из методов, позволяющих учитывать оптимальный план исходной задачи (например, методом последовательного уточнения оценок).

ряда экономических и военных задач отличаются от условий транспортной задачи небольшим числом дополнительных линейных неравенств. При решении подобных задач естественно отвлечься сначала от условий, отличающих задачу от транспортной. Оптимальный план транспортной задачи может быть затем использован как исходный пункт для решения исследуемой проблемы.

Любой план сопряженной усеченной задачи является планом задачи, сопряженной к расширенной.

Следовательно, оптимальный план сопряженной усеченной задачи может быть использован как начальный план для расширенной сопряженной задачи. Различные алгоритмы, основанные на этом принципе, позволяют сократить трудоемкость расчетов при решении задач линейного программирования за счет использования оптимальных планов уже решенных задач. Число итераций, необходимое для решения исследуемой проблемы, существенно зависит от ее близости к уже решенной задаче. Здесь под близкими задачами подразумеваются задачи, отличающиеся несколькими условиями или несколькими коэффициентами линейной формы.

Если две задачи линейного программирования различаются лишь небольшими флюктуациями линейной формы, то оптимальный план одной из них целесообразно принимать в качестве исходного при решении другой. В подобных случаях естественно пользоваться методами, связанными с решением прямой задачи (например, методом последовательного улучшения плана).

В ряде задач от искомым переменных требуется не только неотрицательность, но и ограниченность каждого или некоторых из них. Такова, например, транспортная задача с ограниченными пропускными способностями в отдельных направлениях или задача планирования производства и потребления при ограниченных возможностях расширения производства. Дополнительные условия $x_i \leq d_i$ увеличивают число ограничений, что при использовании общих методов линейного программирования ведет к существенному повышению трудоемкости расчета. Однако относительно несложные естественные усовершенствования алгоритмов позволяют учитывать подобные условия так же, как учитывается

неотрицательность переменных. При большом количестве ограниченных переменных можно таким образом существенно упростить вычисления.

Известны и другие приемы, упрощающие применение общих методов линейного программирования при специальном виде ограничений. Особый интерес для приложений представляет так называемое блочное программирование.

В задачах многоэтапного планирования использование ресурсов на последующем этапе обуславливается расходами на предыдущих этапах. Продукция, полученная на первых этапах, увеличивает общее количество ресурсов и может расходоваться в дальнейшем. Поэтому при сведении подобных задач к схемам линейного программирования часто приходится накладывать ограничения на расходы первого этапа, первых двух этапов, первых трех этапов и т. д. Матрица условий в таких случаях имеет вид

$$A = \begin{bmatrix} A_{11} & 0 & \dots & 0 \\ A_{21} & A_{22} & \dots & 0 \\ \cdot & \cdot & \cdot & \cdot \\ A_{k1} & A_{k2} & \dots & A_{kk} \end{bmatrix},$$

где $A_{ij} = \|A_{ij}\|_{m_i \times n_j}$ — матрицы-блоки.

Блочно-треугольная форма матрицы A позволяет упростить алгоритм вычислений при решении задачи по методу последовательного улучшения плана. Изложение соответствующих рекомендаций можно найти в [33].

§ 2. ЛИНЕЙНОЕ ПРОГРАММИРОВАНИЕ И ТЕОРИЯ ИГР

2.1. В § 9 гл. 1 и § 14 гл. 2 уже отмечалась связь линейного программирования и теории игр. В гл. 1 среди методов решения задач линейного программирования упоминается метод «вилки», предложенный Брауном для приближенного решения прямоугольных игр. В гл. 2 игровая задача сведена к задаче линейного программирования. Те же рассуждения, что и в рассмотренном примере, служат основанием для сведения любой прямоугольной игры к задаче линейного программирования. Можно показать и обратное — любая задача линейного программирования может быть сведена к решению некоторой прямоугольной игры с нулевой

суммой. Таким образом, методы теории игр могут быть использованы в линейном программировании, равно как и алгоритмы линейного программирования могут оказаться полезными при решении игр.

Напомним некоторые определения теории игр и укажем методы решения игр, развитие которых может привести к удобным алгоритмам для широкого круга задач линейного программирования.

Прямоугольная или матричная игра двух лиц с нулевой суммой определяется следующим образом.

1. Имеется два игрока. Игра одноходовая. У первого игрока m возможностей (m чистых стратегий), а второму предоставляется выбор из n вариантов (n чистых стратегий).

2. Каждый игрок выбирает свою стратегию, не зная выбора противника.

3. Если I игрок выберет i -ю чистую стратегию, а II — j -ю, то в результате игры I выиграет величину a_{ij} , II — проиграет a_{ij} (т. е. выигрыш II игрока равен $-a_{ij}$).

Матрица

$$\|a_{ij}\| = \begin{vmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \cdot & \cdot & \cdot & \cdot \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{vmatrix}$$

называется платежной матрицей игры. Номера строк платежной матрицы определяются чистыми стратегиями I игрока, номера столбцов — чистыми стратегиями второго.

4. Если I игрок с вероятностью x_1 выбирает свою первую чистую стратегию, с вероятностью x_2 — вторую чистую стратегию и т. д. $\left(x_i \geq 0, \sum_{i=1}^m x_i = 1\right)$, а II игрок с вероятностью y_j выбирает свою j -ю чистую стратегию $\left(y_j \geq 0, \sum_{j=1}^n y_j = 1\right)$, то средний платеж первому игроку равен

$$M(X, Y) = \sum_{j=1}^n \sum_{i=1}^m a_{ij} x_i y_j$$

и соответственно выигрыш II равен $-M$.

5. Набор чисел $X = (x_1, x_2, \dots, x_m)$ ($x_i \geq 0$, $\sum x_i = 1$), определяющий вероятности, с которыми используются чистые стратегии первого игрока, называется его смешанной стратегией. Аналогично определяется смешанная стратегия II игрока. Очевидно, смешанная стратегия, все компоненты которой, кроме одной, равны нулю, является чистой стратегией.

6. Цель I игрока выбрать стратегию, обеспечивающую ему максимальную плату. Цель II игрока выбрать стратегию, которая сводит выигрыш противника к минимуму.

7. Основная теорема теории игр утверждает, что во всякой парной матричной игре с нулевой суммой существуют оптимальные смешанные стратегии для обоих партнеров. Гарантированный выигрыш I игрока (и, следовательно, гарантированный проигрыш II игрока) называется ценой игры и равен

$$v = \min_Y \max_X M(X, Y) = \max_X \min_Y M(X, Y).$$

Вычисление оптимальных стратегий противников и цены игры называется решением игры.

8. Если X^0 , Y^0 — оптимальные стратегии, а X , Y — произвольные стратегии игроков, то

$$M(X, Y^0) \leq M(X^0, Y^0) = v \leq M(X^0, Y).$$

2.2. Пример сведения прямоугольной игры с нулевой суммой к задаче линейного программирования, рассмотренный в § 14 гл. 2, носит достаточно общий характер. Здесь мы покажем, как решается обратная задача.

Рассмотрим пару двойственных задач линейного программирования.

Сформулируем прямую задачу следующим образом. Требуется обратить в минимум линейную форму

$$L = \sum_{j=1}^n c_j x_j$$

при условиях

$$\left. \begin{aligned} \sum_{j=1}^n a_{ij} x_j &\geq b_i \quad (i=1, 2, \dots, m) \\ x_j &\geq 0 \quad (j=1, 2, \dots, n) \end{aligned} \right\}. \quad (7.5)$$

Этой задаче соответствует следующая сопряженная задача. Требуется вычислить максимум линейной формы

$$L_j = \sum_{i=1}^m b_i y_i$$

при условиях

$$\left. \begin{aligned} \sum_{i=1}^m a_{ij} y_i &\leq c_j \quad (j=1, 2, \dots, n) \\ y_i &\geq 0 \quad (i=1, 2, \dots, m) \end{aligned} \right\}. \tag{7.6}$$

Покажем, что оптимальные планы прямой и сопряженной задач линейного программирования совпадают с решением прямоугольной игры двух лиц с нулевой суммой и со следующей платежной матрицей:

$$A = \left\| \begin{array}{ccccccc} & & & & -a_{11} & -a_{12} \dots -a_{1n} & b_1 \\ & & & & -a_{21} & -a_{22} \dots -a_{2n} & b_2 \\ & & 0 & & \dots & \dots & \dots \\ & & & & -a_{m1} & -a_{m2} \dots -a_{mn} & b_m \\ a_{11} & a_{21} \dots & a_{m1} & & & & -c_1 \\ a_{12} & a_{22} \dots & a_{m2} & & & & -c_2 \\ \dots & \dots & \dots & & 0 & & \dots \\ a_{1n} & a_{2n} \dots & a_{mn} & & & & -c_n \\ -b_1 & -b_2 \dots -b_m & c_1 & c_2 & \dots & c_n & 0 \end{array} \right\|.$$

Матрица A кососимметрична, поэтому цена игры v равна нулю. Действительно, если бы у I игрока имелась стратегия, гарантирующая ему по меньшей мере $v > 0$, то, применяя соответствующую стратегию, II игрок мог бы свести платеж первому самое большее к $(-v)$. Мы получили бы противоречие.

Обозначим оптимальную стратегию второго игрока через Z

$$Z = (z_1, z_2, \dots, z_m, z_{m+1}, \dots, z_{m+n}, z_{m+n+1}),$$

$$z_k \geq 0, \quad \sum_{k=1}^{m+n+1} z_k = 1.$$

Какую бы чистую стратегию не применил I игрок, его партнер, придерживаясь своей оптимальной сме-

шанной стратегии, не позволит первому получить положительную плату ($v=0$). Т. е.

$$-(a_{i1}z_{m+1} + \dots + a_{in}z_{m+n}) + b_i z_{m+n+1} \leq 0$$

$$(i=1, 2, \dots, m); \quad (7.7)$$

$$(a_{1j}z_1 + \dots + a_{mj}z_m) - c_j z_{m+n+1} \leq 0$$

$$(j=1, 2, \dots, n); \quad (7.8)$$

$$-(b_1z_1 + \dots + b_mz_m) + (c_1z_{m+1} + \dots + c_nz_{m+n}) \leq 0. \quad (7.9)$$

Пусть в оптимальной стратегии II игрока $z_{m+n+1} > 0$ (можно показать, что если это не так, то соответствующие задачи линейного программирования неразрешимы).

Разделим все члены неравенств (7.7) — (7.9) на z_{m+n+1} и обозначим

$$\frac{z_{m+j}}{z_{m+n+1}} = x_j, \quad \frac{z_i}{z_{m+n+1}} = y_i$$

$$(j=1, 2, \dots, n; \quad i=1, 2, \dots, m).$$

В новых обозначениях условия (7.7) — (7.9) принимают вид

$$a_{i1}x_1 + \dots + a_{in}x_n \geq b_i \quad (i=1, 2, \dots, m); \quad (7.10)$$

$$a_{1j}y_1 + \dots + a_{mj}y_m \leq c_j \quad (j=1, 2, \dots, n); \quad (7.11)$$

$$c_1x_1 + \dots + c_nx_n \leq b_1y_1 + \dots + b_my_m; \quad (7.12)$$

$$x_j \geq 0, \quad y_i \geq 0. \quad (7.13)$$

Из условий (7.10) — (7.13) следует, что $X = (x_1, \dots, x_n)$ является оптимальным планом прямой задачи, а $Y = (y_1, \dots, y_m)$ оказывается решением сопряженной задачи. Пусть это неверно. Тогда существует набор неотрицательных чисел $\tilde{X} = (\tilde{x}_1, \dots, \tilde{x}_n)$, которые удовлетворяют условиям (7.5),

$$a_{i1}\tilde{x}_1 + \dots + a_{in}\tilde{x}_n \geq b_i,$$

$$\tilde{x}_j \geq 0 \quad (7.14)$$

и придает линейной форме L значение, меньшее чем $L(X)$, т. е.

$$L(\tilde{X}) = \sum_{j=1}^n c_j \tilde{x}_j < \sum_{j=1}^n c_j x_j = L(X). \quad (7.15)$$

Умножим каждое из неравенств (7.14) на y_i и просуммируем по i . Затем умножим каждое из неравенств (7.11) на $\tilde{x}_j$ и просуммируем по j . Получим

$$\sum_{i=1}^m b_i y_i \leq \sum_{i=1}^m \left(\sum_{j=1}^n a_{ij} \tilde{x}_j \right) y_i = \sum_{j=1}^n \left(\sum_{i=1}^m a_{ij} y_i \right) \tilde{x}_j \leq \sum_{j=1}^n c_j \tilde{x}_j.$$

Полученное неравенство противоречит условиям (7.12) и (7.15). Точно так же доказывается, что $Y = (y_1, \dots, y_m)$ является решением сопряженной задачи.

Мы привели решение пары двойственных задач линейного программирования к эквивалентной симметричной игре с платежной матрицей порядка $(m+n+1)$.

В случае, когда все компоненты векторов условий и векторов ограничений прямой и сопряженной задач положительны, можно построить игру с платежной матрицей из n строк и m столбцов, оптимальные стратегии которой определяют решения двойственной пары задач линейного программирования. Платежная матрица эквивалентной игры имеет вид

$$\tilde{A} = \left\| \begin{array}{cccc} \frac{a_{11}}{b_1 c_1} & \frac{a_{21}}{b_2 c_1} & \cdots & \frac{a_{m1}}{b_m c_1} \\ \frac{a_{12}}{b_1 c_2} & \frac{a_{22}}{b_2 c_2} & \cdots & \frac{a_{m2}}{b_m c_2} \\ \cdot & \cdot & \cdot & \cdot \\ \frac{a_{1n}}{b_1 c_n} & \frac{a_{2n}}{b_2 c_n} & \cdots & \frac{a_{mn}}{b_m c_n} \end{array} \right\|.$$

Если $U = (u_1, \dots, u_n)$ — оптимальная стратегия I игрока, $W = (w_1, \dots, w_m)$ — оптимальная стратегия II игрока, а v — цена игры, то наборы чисел

$$x_j = \frac{u_j}{c_j v} \text{ и } y_i = \frac{w_i}{b_i v}$$

$$(j = 1, 2, \dots, n; \quad i = 1, 2, \dots, m)$$

являются соответственно решением прямой и сопряженной задач линейного программирования. Доказательство этого предложения предоставляем читателю.

Порядок платежной матрицы во втором представлении задач линейного программирования в виде эквивалентной игры ниже, чем в первом представлении. Тем не менее, в ряде случаев удобней пользоваться первым представлением, так как для некоторых вычислительных методов выгодней иметь игру в симметричной форме.

2.3. В последние годы появилось значительное количество работ, содержащих различные методы решения игр. Известные в теории игр формулы для выделения оптимальных стратегий и вычисления цены игры (например, [59]) фактически сводятся к вычислению координат всех вершин многогранников условий соответствующих задач линейного программирования. При этом не приводится правил, которые позволили бы упорядочить перебор вершин. Ясно, что при сколь-нибудь значительном количестве чистых стратегий неупорядоченный перебор — чрезмерно трудоемкая работа.

Итерационный метод Брауна (метод «вилки»), упомянутый в § 9 гл. 1, может быть использован для приближенного решения игр с большим количеством чистых стратегий и задач линейного программирования со значительным числом переменных. Однако получение высокой точности решения требует, как правило, весьма большого количества итераций по методу «вилки». По-видимому, медленная сходимость метода «вилки» связана с тем, что на каждом шагу все предыдущие выборы противников учитываются с одинаковым весом. Следует полагать, что рациональный подбор весов позволит существенно ускорить сходимость метода. Но метод, обладающий конечной памятью, т. е. учитывающий лишь фиксированное число предшествующих выборов, вообще говоря, не пригоден для всех игр. Ниже приводится краткая характеристика еще двух методов решения игр. Эти методы могут быть использованы для приближенного вычисления оптимальных планов задач линейного программирования.

Метод решения игр посредством дифференциальных уравнений применим к симметричным играм. Идея метода в следующем. Цена симметричной игры, как мы

уже видели, равна нулю. Начнем приближение к оптимальной стратегии II игрока с некоторой стратегии

$$Y(0) = \{y_1(0), \dots, y_n(0)\}.$$

Если $Y(0)$ не сводит платеж I игроку самое большое к нулю, к стратегии прикладывается сила (подлежащая формальному описанию системой дифференциальных уравнений), которая стремится приблизить ее к состоянию равновесия.

Браун и фон Нейман [24] в 1950 г. предложили следующую систему уравнений для вычисления $Y(t) = \{y_1(t), \dots, y_n(t)\}$

$$\frac{dy_i(t)}{dt} = \varphi_i(Y) - \varphi(Y) y_i(t), \quad (7.16)$$

$$(i = 1, 2, \dots, n),$$

где

$$\varphi(Y) = \sum_{i=1}^n \varphi_i(Y),$$

а $\varphi_i(Y) \geq 0$ равна выигрышу I игрока, выбравшего свою i -ю чистую стратегию против смешанной стратегии $Y(t)$ II игрока. Если выигрыш I игрока окажется при этом отрицательным, $\varphi_i(Y)$ предполагается равной нулю.

Система дифференциальных уравнений определяет $y_i(t) \geq 0$ для всех i и t и обеспечивает равенство

$$\sum_{i=1}^n y_i(t) = 1$$

для всех t . Это значит, что при любом t $Y(t)$ определяет смешанную стратегию II игрока. Кроме того, можно показать, что

$$\varphi_i(Y) \leq \frac{c}{1 + c_i},$$

где

$$c = \sqrt{\sum_{i=1}^n \varphi_i^2[Y(0)]},$$

т. е. с течением времени II игрок ограничивает выигрыш партнера все меньшим числом — в пределе ценой

игры — нулем. Число s характеризует качество начального приближения $Y(0)$.

Рассмотрим еще один метод решения прямоугольных игр [61], который при доработке обещает стать эффективным методом приближенного решения задач линейного программирования.

Идея метода в следующем. Начнем с произвольных стратегий игроков

$$X' = (x_1', \dots, x_m'), \quad Y' = (y_1', \dots, y_n').$$

$$\left(\text{Естественно, } x_i' \geq 0, \quad y_j' \geq 0, \quad \sum_{i=1}^m x_i' = \sum_{j=1}^n y_j' = 1. \right)$$

При выбранных стратегиях выигрыш I игрока равен

$$v' = \sum_{j=1}^n \sum_{i=1}^m a_{ij} x_i' y_j'.$$

Далее указывается способ, позволяющий шаг за шагом совершенствовать стратегии противников. В результате h -го шага получаем

$$X^{(h)} = (x_1^{(h)}, \dots, x_m^{(h)}), \quad Y^{(h)} = (y_1^{(h)}, \dots, y_n^{(h)}), \quad v^{(h)}.$$

Качество h -го приближения характеризуется показателем

$$\varphi^{(h)} = \sum_j' \left(v^{(h)} - \sum_{i=1}^m a_{ij} x_i^{(h)} \right)^2 + \sum_i' \left(\sum_{j=1}^n a_{ij} y_j^{(h)} - v^{(h)} \right)^2.$$

Сумма $\sum_j'$ берется по всем j , для которых $\sum_{i=1}^m a_{ij} x_i^{(h)} \leq v^{(h)}$.

Сумма $\sum_i'$ берется по индексам i , для которых

$\sum_{j=1}^n a_{ij} y_j \geq v^{(h)}$. Последовательное совершенствование стратегий производится следующим образом.

Вводятся параметры w_j' и u_i' :

$$w_j^{(h)} = \begin{cases} 0, & \text{если } \sum_{i=1}^m a_{ij} x_i^{(h)} \geq v^{(h)}, \\ v^{(h)} - \sum_{i=1}^m a_{ij} x_i^{(h)}, & \text{если } \sum_{i=1}^m a_{ij} x_i^{(h)} \leq v^{(h)}; \end{cases}$$

$$u_i^{(h)} = \begin{cases} 0, & \text{если } \sum_{j=1}^n a_{ij} y_j^{(h)} \leq v^{(h)}, \\ \sum_{j=1}^n a_{ij} y_j^{(h)} - v^{(h)}, & \text{если } \sum_{j=1}^n a_{ij} y_j^{(h)} \geq v^{(h)}. \end{cases}$$

Система чисел $w_j^{(h)}$ и $u_i^{(h)}$ служит для вычисления

$$\tilde{x}_i^{(h)} = \frac{u_i^{(h)}}{\sum_{k=1}^m u_k^{(h)}}, \quad \tilde{y}_j^{(h)} = \frac{w_j^{(h)}}{\sum_{k=1}^n w_k^{(h)}},$$

$$\tilde{v}^{(h)} = \sum_{j=1}^n \sum_{i=1}^m a_{ij} \tilde{x}_i^{(h)} \tilde{y}_j^{(h)}.$$

Очередное приближение

$$X^{(h+1)} = (x_1^{(h+1)}, \dots, x_m^{(h+1)}),$$

$$Y^{(h+1)} = (y_1^{(h+1)}, \dots, y_n^{(h+1)}), \quad v^{(h+1)}$$

вычисляется как средневзвешенное величин $X^{(h)}$, $Y^{(h)}$, $v^{(h)}$ и $\tilde{X}^{(h)}$, $\tilde{Y}^{(h)}$, $\tilde{v}^{(h)}$, причем веса выбираются так, чтобы показатель $\varphi^{(h)}$ был возможно меньше.

Можно показать, что

$$\varphi^{(1)} \geq \varphi^{(2)} \geq \dots \geq \varphi^{(h)} \dots$$

и

$$\varphi^{(h)} \leq \frac{m+n}{h} \left[\max_{i,j} a_{ij} - \min_{i,j} a_{ij} \right].$$

Фон Нейман сравнивает по трудоемкости предложенный им метод решения игр и I алгоритм метода последовательного улучшения плана. При этом он ука-

зывает, что априорные оценки метода последовательного улучшения плана, вообще говоря, менее благоприятны, чем для описанного итеративного метода решения игр. Однако опытные оценки числа итераций в методе последовательного улучшения плана, полученные в результате решения многочисленных практических задач, оказались значительно лучше пределов, гарантируемых теорией. Фон Нейман указывает также идеи различных приемов, которые, видимо, ускорят сходимость предложенного им метода. Эти приемы сводятся к сглаживанию итеративной последовательности. Каждый последующий шаг ставится в определенную зависимость от нескольких предшествующих итераций. Следует полагать, что такой прием позволит так же ускорить сходимость метода «вилки».

В заключение параграфа заметим, что каждый из алгоритмов линейного программирования можно переформулировать в терминах игровой задачи и использовать для ее решения.

§ 3. ПЕРСПЕКТИВНЫЕ ВОПРОСЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Линейное программирование бурно развивается во многих направлениях. Некоторые из этих направлений, несмотря на ограниченность результатов в настоящее время, представляются весьма перспективными. Мы остановимся здесь только на двух таких вопросах.

3.1. Параметрическое программирование. При постановке задач планирования народного хозяйства и управления военной техникой существенные затруднения вызывает определение постоянных, входящих в критерий качества и ограничения. Неполная информация об условиях задачи и возможность изменения условий в некоторых пределах не позволяют находить точные значения необходимых параметров. Чаше удается указать диапазон, в котором они могут находиться. В связи с этим становится актуальным исследование влияния вариаций коэффициентов линейной формы и ограничений на оптимальность плана задачи линейного программирования. Параметрическое программирование, предложенное С. Гассом [45], является первым шагом в этом направлении.

Параметрическое программирование позволяет решать следующую задачу.

Необходимо вычислить планы

$$X^{(\lambda)} = (x_1^{(\lambda)}, x_2^{(\lambda)}, \dots, x_n^{(\lambda)}),$$

обращающие в максимум (минимум) линейные формы

$$L^{(\lambda)} = \sum_{j=1}^n (c_j^{(1)} + \lambda c_j^{(2)}) x_j$$

при условиях

$$\sum_{j=1}^n a_{ij} x_j = b_i, \quad i = 1, 2, \dots, m,$$

$$x_j \geq 0, \quad j = 1, 2, \dots, n,$$

для всех λ в диапазоне $[\lambda_1, \lambda_2]$.

Естественный путь — решение большого числа задач линейного программирования для фиксированных значений λ в диапазоне $[\lambda_1, \lambda_2]$ — чрезвычайно трудоемко. Гасс показал, как можно, несколько усовершенствовав общие методы линейного программирования, получить решение задачи для всего диапазона изменения λ . При этом общее количество итераций, как правило, незначительно превышает число шагов, необходимых для решения задачи при фиксированном λ .

Идею метода легко пояснить, используя наглядную геометрическую интерпретацию задач линейного программирования.

Мы видели, что области определения линейной формы соответствует в пространстве переменных $x_1, x_2, \dots, x_n$ выпуклый многогранник. Коэффициенты линейной формы $L^{(\lambda)}(X)$ определяют при фиксированном λ (в частности, при $\lambda = \lambda_1$) в пространстве $(x_1, \dots, x_n)$ семейство параллельных гиперплоскостей. Зафиксируем одну из этих гиперплоскостей. Пусть она пересекает выпуклый многогранник. Чтобы решить задачу на максимум $L^{(\lambda_1)}(X)$, необходимо, сохраняя направляющие косинусы гиперплоскости, двигать ее в направлении возрастания $L^{(\lambda_1)}$ до тех пор, пока она не пройдет через крайнюю вершину A многогранника условий. Будем на-

зывать полученную таким образом гиперплоскость экстремальной. Ясно, что экстремальную гиперплоскость можно покачивать в определенном диапазоне направлений относительно принадлежащей ей вершины многогранника так, чтобы она при этом не пересекала многогранника. Покачиванию гиперплоскости соответствует изменение коэффициентов линейной формы. В частности, изменяя λ_1 в диапазоне $\lambda' \leq \lambda_1 \leq \lambda''$, можно изменять направление экстремальной гиперплоскости, не пересекая при этом многогранника. Это значит, что оптимальный план, вычисленный для значения параметра λ , равного λ_1 , обращает в максимум $L^{(\lambda)}$ во всем диапазоне $(\lambda' \leq \lambda \leq \lambda'')$. Значение параметра λ'' соответствует гиперплоскости, которая, помимо вершины A , содержит еще одну (соседнюю) вершину (B) многогранника условий. Величина λ'' , соответствующая вершине B , указывает, какой вектор следует ввести в базис для того, чтобы получить решение задачи линейного программирования для следующего интервала изменения λ . Покачивая теперь гиперплоскость относительно вершины B , определяем новое множество гиперплоскостей, или, что то же самое, новый диапазон изменения $(\lambda'' \leq \lambda \leq \lambda''')$, для которого вершина B является оптимальной. Так следует переходить от одного интервала изменения λ к соседнему (от одной вершины многогранника условий к соседней), пока не будет исчерпан определяемый задачей диапазон изменения параметра. Переход от одного интервала изменения λ к следующему связан с одной дополнительной итерацией в любом общем методе линейного программирования.

Гасс утверждает [45], что им была решена таким образом задача линейного программирования, содержащая 65 переменных и 33 условия. При этом коэффициенты линейной формы линейно зависели от параметра λ , который мог изменяться вдоль всей положительной полуоси $(0 \leq \lambda < \infty)$. В процессе решения диапазон изменения λ был расчленен на 23 интервала, в каждом из которых оптимальный план оставался неизменным. Задача была решена для всех положительных λ за 53 итерации.

Пример Гасса свидетельствует об эффективности параметрического программирования.

В сопряженной задаче линейного программирования коэффициенты линейной формы исходной задачи переходят в компоненты вектора ограничений. Поэтому параметрическое программирование может быть использовано и для решения задач, в которых ограничения линейно зависят от параметра. Этот случай хорошо иллюстрируется с помощью второй геометрической интерпретации решения задачи линейного программирования (§ 7 гл. 1).

Экстремум линейной формы соответствует крайней точке A пересечения прямой $\mathcal{L}$ и выпуклого многогранного конуса. Пусть точка A принадлежит грани H конуса (задача по условию невырожденная). При изменении параметра λ меняются компоненты вектора ограничений и прямая $\mathcal{L}$ смещается, оставаясь параллельной своему первоначальному направлению. Смещается и точка A пересечения оси $\mathcal{L}$ и конуса. Диапазон изменения λ , при котором точка A не выходит за пределы грани H , соответствует одному и тому же базису оптимального плана задачи линейного программирования. Когда в процессе изменения λ точка A перейдет на соседнюю грань конуса, получим новое решение, соответствующее очередному интервалу изменения параметра. Переход к соседней грани отвечает дополнительной итерации и приводит к новому базису.

Рассмотренная задача — относительно частный случай параметрического программирования. Здесь коэффициенты линейной формы зависят только от одного параметра и зависимость линейная. Для приложений представляют интерес всесторонние обобщения этой задачи.

3.2. Линейное программирование в условиях неопределенности. Проблемы линейного программирования в условиях неопределенности вызваны к жизни теми же факторами, что и параметрическое программирование. Далеко не во всех случаях планирующим организациям известны все необходимые данные для планирования хозяйства или управления боем. Однако опыт, статистика и исследование процессов, определяющих изменение исходных данных для планирования, часто позволяют устанавливать те или иные вероятностные характеристики коэффициентов линейной формы, векторов

условий и вектора ограничений рассматриваемой задачи линейного программирования.

В таких случаях говорят о задачах линейного программирования в условиях неопределенности или о линейном программировании при неполной информации. Этот круг вопросов в настоящее время интенсивно разрабатывается. Исследуются специальные методы решения подобных задач и в то же время изучаются возможности сведения линейного программирования при неполной информации к обычным задачам линейного программирования.

Мы ограничимся здесь рассмотрением примера сведения задачи управления в условиях неопределенности к задаче линейного программирования при полной информации [65].

В § 10 гл. 2 была рассмотрена задача о рациональной организации перевозок угля из n пунктов производства в m пунктов потребления топлива. При этом производство и спрос в каждом пункте на каждом этапе был строго фиксирован. Введем теперь в задачу элемент неопределенности. Пусть производство угля в каждом пункте по-прежнему фиксировано, а спрос от этапа к этапу меняется по различным причинам, причем к моменту планирования перевозок нет полной информации о потребностях каждого пункта на очередной этап. Проанализировав размеры потребления на предыдущих этапах и причины, определяющие возможные колебания спроса, можно составить для каждого пункта потребления функцию распределения спроса $f_j(b_j)$ на очередном этапе.

Пусть в j -й пункт направлено y_j тонн угля. Ясно, что количество угля, которое j -й пункт потребляет на этом этапе, равно меньшему из двух чисел b_j и y_j , где b_j — его истинная потребность, а y_j — предполагаемый спрос, в соответствии с которым пункт снабжается углем. Поэтому математическое ожидание потребления угля в j -м пункте можно записать в виде

$$\int_0^{y_j} b f_j(b) db + y_j \int_{y_j}^{\infty} f_j(b) db.$$

Пусть реализация одной тонны угля в j -м пункте потребления приносит доход, равный g_j рублей. Тогда

ожидаемый доход от реализации угля, направленного в j -й пункт, равен

$$\varphi_j(y_j) = g_j \left[\int_0^{y_j} b f_j(b) db + y_j \int_{y_j}^{\infty} f_j(b) db \right], \quad (7.17)$$

а математическое ожидание дохода от реализации угля во всех n пунктах потребления равно

$$\sum_{j=1}^n \varphi_j(y_j).$$

Сохраняя обозначения § 10 гл. 2, запишем выражение для расходов на перевозки угля из m пунктов производства в n пунктов спроса

$$L = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij}.$$

Задача планирования перевозок в условиях неопределенности сводится таким образом к вычислению такой системы неотрицательных чисел x_{ij} (количество угля, перевозимого из i -го пункта производства в j -й пункт потребления) и y_j (общее количество угля, направляемого в j -й пункт потребления), чтобы превышение доходов над расходами было максимально возможным. Другими словами, необходимо обратить в минимум

$$R = \sum_{j=1}^n \sum_{i=1}^m c_{ij} x_{ij} - \sum_{j=1}^n \varphi_j(y_j).$$

При этом, как следует из рассуждений § 10 гл. 2, должны соблюдаться условия

$$\sum_{j=1}^n x_{ij} = a_i,$$

$$\sum_{i=1}^m x_{ij} = y_j,$$

$$\sum_{j=1}^n y_j = \sum_{i=1}^m a_i,$$

$$x_{ij} \geq 0, \quad y_j \geq 0 \quad (i=1, 2, \dots, m; j=1, 2, \dots, n).$$

Заметим, что функции $\{-\varphi_j(y_j)\}$ выпуклы. Действительно, дифференцируя выражение (7.17) и используя правило дифференцирования интеграла с пределами, зависящими от параметра, получим

$$\begin{aligned} -\varphi_j'(y_j) &= -g_j \left[y_j \varphi_j(y_j) + \int_{y_j}^{\infty} f_j(b) db - y_j \varphi_j(y_j) \right] = \\ &= -g_j \int_{y_j}^{\infty} f_j(b) db. \end{aligned}$$

Функция $f_j(b)$, как функция распределения, неотрицательна. Поэтому $\int_{y_j}^{\infty} f_j(b) db$ монотонно убывает с увеличением y_j , а $\{-\varphi_j'(y_j)\}$ монотонно возрастает. Следовательно, $\{-\varphi_j(y_j)\}$ является выпуклой функцией.

Мы уже видели, что методы линейного программирования позволяют вычислять с любой заранее заданной точностью минимум линейной комбинации выпуклых функций, если искомые переменные удовлетворяют линейным условиям. Таким образом, рассматриваемая задача планирования в условиях неопределенности свелась к обычной задаче линейного программирования.

ЛИТЕРАТУРА

1. Баранов Г. Г. О выборе допусков, обеспечивающих заданную точность механизма и наименьшую стоимость его изготовления. Труды Института машиноведения, вып. 11, 1956.
2. Глейзал. Алгоритм для решения проблемы транспортировки. «Математика», 1958, 2, 1.
3. Данциг Дж. Б., Форд П. Р., Фулкерсон Д. Р. Алгоритм для одновременного решения прямой и двойственной задач линейного программирования. Сб. «Линейные неравенства и смежные вопросы». Изд-во иностранной литературы, 1959.
4. Даффин Р. Дж. Бесконечные программы. Сб. «Линейные неравенства и смежные вопросы». Изд-во иностранной литературы, 1959.
5. Дунаев П. Ф. Методика расчетов рациональных допусков. «Станки и инструмент», 1952, 6.
6. Канторович Л. В. Математические методы в организации и планировании производства. Изд-во ЛГУ, 1939, стр. 67.
7. Канторович Л. В. Об одном эффективном методе решения некоторых классов экстремальных проблем. ДАН СССР, 1940, 28, 3, стр. 212—215.
8. Канторович Л. В. О перемещении масс. ДАН СССР, 1942, 37, 7—8, стр. 227—229.
9. Канторович Л. В. О методах анализа некоторых экстремальных планово-производственных задач. ДАН СССР, 1957, 115, 3, стр. 441—444.
10. Канторович Л. В., Рубинштейн Г. Ш. Об одном функциональном пространстве и некоторых экстремальных задачах. ДАН СССР, 1957, 115, 6, стр. 1058—1061.
11. Канторович Л. В., Гавурин М. К. Применение математических методов в вопросах анализа грузопотоков. Сб. «Проблемы повышения эффективности работы транспорта». Изд-во АН СССР, 1949, стр. 110—138.
12. Канторович Л. В., Залгаллер В. А. Расчет рационального раскроя промышленных материалов. Лениздат, 1951, стр. 198.
13. Рубинштейн Г. Ш. Задача о крайней точке пересечения оси с многогранником и ее приложение к исследованию конечной системы линейных неравенств. ДАН СССР, 1955, 100, 4, стр. 627—630.
14. Рубинштейн Г. Ш. Обобщение задачи о крайней точке пересечения оси с выпуклым многогранником. ДАН СССР, 1957, 113, 5, стр. 987—990.

15. Соколовский А. П. Курс технологии машиностроения. Машгиз, 1947.
16. Толстой А. Н. Методы устранения нерациональных перевозок при планировании. «Социалистический транспорт», 1939, 9, стр. 28—51.
17. Фадеева В. Н. Вычислительные методы линейной алгебры. Гостехиздат, 1950, стр. 240.
18. Шилов Г. Е. Введение в теорию линейных пространств. Гостехиздат, 1956.
19. Линейные неравенства и смежные вопросы. Сб. статей под ред. Г. У. Куна и А. У. Такера. Изд-во иностранной литературы, 1959.
20. Arrow K. J., Hurwicz L. Gradient methods for constrained maxima. *Operat. Res.*, 1957, 5, 2, p. 258—265.
21. Batchelor J. H. *Bibliography on Operations Research*, St. Louis, 2-nd ed., 1959.
22. Beale E. M. L. An alternative method for linear programming, *Proc. Cambridge Philos. Soc.*, 1954, 50, 4, p. 513—523.
23. Brown G. W. Iterative solution of games by fictitious play, *Activity analysis of production and allocation*, ed. T. C. Koopmans, Cowles Commission Monograph, 13, Wiley, New York, 1951, p. 374—376.
24. Brown G. W., von Neumann J. Solution of games by differential equations, *Contributions to the theory of games*, vol. I, ed. H. W. Kuhn and A. W. Tucker, *Annals of Math. Studies* 24, Princeton, 1950, p. 73—79.
25. Charnes A. Optimality and degeneracy in linear programming, *Econometrica*, 1952, 20, 2, p. 160—170.
26. Charnes A., Cooper W. W., Mellon B. Blending aviation gasolines — a study in programming interdependent activities in an integrated oil company. *Econometrica*, 1952, 20, 2, p. 135—159.
27. Charnes A., Lemke C. E. The minimization of non-linear separable convex functionals, *Naval Res. Logist. Quart.*, 1954, 1, 4, p. 301—312.
28. Dantzig G. B. Programming of interdependent activities, Mathematical model. *Econometrica* 1949, 17, p. 200—211.
29. Dantzig G. B. Maximization of a linear function of variables subject to linear inequalities, *Activity analysis of production and allocation*, ed. T. C. Koopmans, Cowles Commission Monograph, 13, Wiley, New York, 1951, p. 339—347.
30. Dantzig G. B. Application of the simplex method to a transportation problem, *Activity analysis of production and allocation*, ed. T. C. Koopmans, Cowles Commission Monograph, 13, Wiley, New York, 1951, p. 359—373.
31. Dantzig G. B. A proof of the equivalence of the programming problem and the game problem, *Activity analysis of production and allocation*, ed. T. C. Koopmans, Cowles Commission Monograph, 13, Wiley, New York, 1951, p. 330—335.
32. Dantzig G. B., Constructive proof of the min-max theorem. *Pacific J. Math.*, 1956, 6, 1, p. 25—33.
33. Dantzig G. B. Upper bounds, secondary constraints and block triangularity in linear programming. *Econometrica*, 1955, 32, 2, p. 174—183.

34. Dantzig G. B. Note on solving linear programs in integers. *Naval Research Logistic Quarterly*, 1959, 4, 1.
35. Dantzig G. B., Orden A. A duality theorem based on the simplex method, *Symposium on linear inequalities and programming*, ed. A. Orden and L. Goldstein (Project SCCOOP, 10), Directorate of Management Analysis, DCS Comptroller, Headquarters US Air Force, Washington, D. C., 1952, p. 51—55.
36. Dantzig G. B., Orden A., Wolfe P. The generalized simplex method for minimizing a linear form under linear inequality restraints. *Pacific J. Math.*, 1955, 5, 2, p. 183—195. /
37. Dantzig G. B., Orchard-Hays W. The product form for the inverse in the simplex method. *Math., Tables Aids Comp.*, 1954, 8, p. 64—67.
38. Egerváry E. Matrixok kombinatorius tulajdonságairól. *Mat. Fiz. Lapok*, 1931, 38, p. 16—28.
39. Ferguson R. O., Dantzig G. B. Problem of routing aircraft. *Aeron. Engineering Rev.*, 1955, 14, p. 51—55.
40. Ford L. R., Fulkerson D. R. A simple algorithm for finding maximal network flows and an application to the Hitchcock problem. *Canad. J. Math.*, 1957, 9, p. 210—218.
41. Ford L. R., Fulkerson D. R. A primal-dual algorithm for the capacitated Hitchcock problem. *Naval Res. Logist. Quart.*, 1957, 4, 1, p. 45—54.
42. Frisch R. La résolution des problèmes de programme linéaire par le méthode du potentiel logarithmique. *Cahiers sém. économétrie*, 1956, 4, p. 7—20, Discuss. p. 20—23.
43. Gale D. The basic theorems of real linear programming and game theory, *Naval Res. Logist. Quart.*, 1956, 3, p. 193—200.
44. Gale D., Kuhn H. W., Tucker A. W. Linear programming and the theory of games, *Activity analysis of production and allocation*, ed. T. C. Koopmans, Cowles Commission Monograph, 13, Wiley, New York, 1951, p. 317—329.
45. Gass S. L. *Linear programming*. Mc Graw-Hill, New York, 1958, p. 223.
46. Gass S. L., Saaty T. L. The computational algorithm for the parametric objective function. *Naval Res. Logist. Quart.*, 1955, 2, p. 39—45.
47. Gomory R. E., Outline of an algorithm for integer solutions to linear programs. *Bull. Amer. Math. Soc.*, 1958, 64, 5, p. 275—278.
48. Hitchcock F. L. Distribution of a product from several sources to numerous localities. *J. Math. Phys.*, (1941) 20, p. 224—230.
49. Jakobs W. W. The caterer problem. *Naval Res. Logist. Quart.*, 1954, 1, p. 154—165.
50. Jakobs W. W. Military applications of linear programming, *Proceedings of the second symposium in linear programming*, vol. 1 and 2, ed. H. A. Antosiewicz, Directorate of Management Analysis, DCS Comptroller, Headquarters US Air Force, Washington, D. C., 1955, p. 1—27.
51. Kuhn H. W. The Hungarian method for solving the assignment problem. *Naval Res. Logist. Quart.*, 1955, 2, p. 83—97.
52. Kuhn H. W. Variants of the Hungarian method for assignment problems. *Naval Res. Logist. Quart.*, 1956, 3, 4, pp. 253—258.

53. K u h n H. W., T u c k e r A. W. Nonlinear programming Proc. of the Second Berkeley Symposium on Mathematical Statistics and Probability (ed. J. Neymann), Univ. Calif., Berkeley, 1951, p. 481—492.

54. L e m k e G. E. The dual method of solving the linear programming problem. Naval Res. Logist. Quart., 1954, 1, 1, p. 36—47.

55. M a n n e A. S., A target-assignment problem. The Journal of the Operations Research Society of America, 1958, 6, 3, p. 307—466.

56. M a r k o v i t z H. The elimination form of the inverse and its application to linear programming. Manag. Sci., 1957, 3, 3, p. 255—269.

57. M a r k o v i t z H., M a n n e A. S. On the solution of discrete programming problems. Econometrica, 1957, 25, 1, p. 84—110.

58. M a r s c h a k J., M i c k e y M. R. Optimal Weapon Systems. Naval Res. Logist. Quart., 1954, 1 and 2.

59. M c K i n s e y J. C. C., Introduction to the theory of games. Mc Grow-Hill, New York, 1954, p. 371.

60. M u n k r e s J. Algorithms for the assignment and transportation problems. J. Soc. Industr. Appl. Math., 1957, 5, 1, p. 32—38.

61. V o n N e u m a n n J. A numerical method to determine optimum strategy. Naval Res. Logist. Quart., 1954, 1, 2, p. 109—115.

62. O r c h a r d - H a y s W. Evolution of linear programming computing techniques. Manag. Sci., 1958, 4, 2, p. 183—190.

63. O s b o r n H. The problem of continuous programs. Pacif. J. Math., 1956, 6, 4, p. 721—731.

64. R i l e y V., G a s s S. L. Bibliography on linear programming and related techniques. Baltimora, 1958.

65. R a d n e r R. The linear team: an example of linear programming under uncertainty, Proceedings of the second symposium in linear programming, vol. 1 and 2, ed. H. A. Antosiewicz, Directorate of Management Analysis, DCS Comptroller, Headquarters US Air Force, Washington, D. C., 1955, p. 381—396.

66. R o b i n s o n J. An iterative method of solving a game. Ann. Math., 1951, 54, p. 296—301.

67. R o h d e F. V. Bibliography on linear programming. Operat. Res., 1957 5, 1, p. 45—62.

68. T u c k e r A. W. Linear and nonlinear programming. Operat. Res., 1957 5, 2, p. 244—257.

69. W a g n e r H. M. The simplex method for beginners. Operat. Res., 1958 6, 2, p. 190—199.

ОГЛАВЛЕНИЕ

Предисловие	3
Глава 1. Основные понятия линейного программирования	7
§ 1. Предмет линейного программирования	7
п. 1.1.	7
п. 1.2.	8
п. 1.3.	9
§ 2. Задачи линейного программирования	11
п. 2.1. Организация снабжения	11
п. 2.2. Выбор системы вооружения	18
п. 2.3. Размещение заказов и загрузка оборудования	20
§ 3. Каноническая форма задач линейного программирования	23
п. 3.1.	23
п. 3.2.	25
§ 4. Геометрический смысл простейших задач линейного программирования	27
п. 4.1.	27
п. 4.2.	29
п. 4.3.	33
п. 4.4.	36
§ 5. Выпуклые многогранники и линейное программирование	38
п. 5.1.	38
п. 5.2.	40
п. 5.3.	45
§ 6. Векторы условий и вектор ограничений	48
п. 6.1.	48
п. 6.2.	52
§ 7. Геометрические интерпретации общей задачи линейного программирования	52
п. 7.1.	52
п. 7.2.	54
п. 7.3.	57
§ 8. Экономическая интерпретация и терминология задачи линейного программирования	59
п. 8.1.	59
п. 8.2.	60
п. 8.3.	61

§ 9. Общая характеристика методов линейного программирования	63
п. 9.1.	63
п. 9.2.	65
п. 9.3.	67
п. 9.4.	70
п. 9.5.	72
п. 9.6.	74
§ 10. Краткая историческая справка	
Глава 2. Практические задачи линейного программирования	75
§ 1. Задача о смеси	78
§ 2. Об оптимальном раскрое материалов	78
§ 3. Распределение самолетов между воздушными линиями	83
§ 4. Сельскохозяйственные задачи	84
§ 5. Задачи о размещении оборудования	88
§ 6. Общая планово-производственная задача	91
§ 7. Проблема составления графиков	95
§ 8. Дилемма: быстро, но дорого, или медленно, но дешево	100
§ 9. Выбор рациональной системы допусков	103
§ 10. Планирование производства и перевозок	110
§ 11. Военные приложения методов линейного программирования (по материалам зарубежной печати)	119
§ 12. Проблема узких мест	120
§ 13. Задача целераспределения	125
§ 14. Теоретико-игровые модели задач линейного программирования	131
Глава 3. Метод последовательного улучшения плана	136
§ 1. Основы метода	137
п. 1.1.	137
п. 1.2.	140
п. 1.3.	145
п. 1.4.	146
§ 2. Выбор начального опорного плана	147
п. 2.1.	147
п. 2.2.	149
п. 2.3.	153
§ 3. Связь между параметрами последовательных приближений	154
п. 3.1.	154
п. 3.2.	157
п. 3.3.	158
п. 3.4.	160
§ 4. I алгоритм метода последовательного улучшения плана	160
п. 4.1.	160
§ 5. Вторая форма критерия оптимальности	178
п. 5.1.	178
п. 5.2.	180
п. 5.3.	181
§ 6. II алгоритм	182
п. 6.1.	182

п. 6.2.	188
п. 6.3.	197
§ 7. Вырожденность	204
п. 7.1.	204
п. 7.2.	207
п. 7.3.	211
§ 8. Исследование общих проблем линейного программирования с помощью метода последовательного улучшения плана	214
Глава 4. Общие методы линейного программирования, основанные на принципе двойственности	217
§ 1. Основы теории двойственности	217
п. 1.1. Постановка вопроса	217
п. 1.2. Теоремы двойственности	222
п. 1.3. Задачи линейного программирования со смешанными условиями	228
п. 1.4. Задачи линейного программирования в канонической форме	234
п. 1.5. Критерии оптимальности	237
п. 1.6. Геометрическая интерпретация задач двойственной пары	242
§ 2. Метод последовательного уточнения оценок	246
п. 2.1. Предварительные замечания	246
п. 2.2. Основы метода	248
п. 2.3. Геометрическая интерпретация метода	258
п. 2.4. Алгоритм метода	259
п. 2.5. Способы отыскания первого приближения	266
п. 2.6. Случай вырожденности	276
п. 2.7. Методы улучшения плана и уточнения оценок	285
п. 2.8. Пример	290
§ 3. Метод последовательного сокращения невязок	300
п. 3.1. Предварительные замечания	300
п. 3.2. Описание метода	303
п. 3.3. Алгоритм метода	310
п. 3.4. Пример	317
п. 3.5. Об оценках быстроты сходимости метода сокращения невязок	323
Глава 5. Транспортная задача	326
§ 1. Постановка вопроса и предварительные замечания	327
п. 1.1. Постановка задачи	327
п. 1.2. О разрешимости транспортной проблемы	333
п. 1.3. Алгоритм построения плана	335
п. 1.4. Транспортная задача с нарушенным балансом производства и потребления	338
п. 1.5. Свойства решений транспортной задачи	340
п. 1.6. Условия невырожденности транспортной задачи	343
п. 1.7. Q задачах, сводящихся к транспортной	346
§ 2. Метод потенциалов	349
п. 2.1. Предварительные замечания	349
п. 2.2. Описание метода потенциалов	350
п. 2.3. Описание алгоритма	356

п. 2.4. Метод минимального элемента	365
п. 2.5. Распространение метода на вырожденные задачи	369
§ 3. Венгерский метод	377
п. 3.1. Предварительные замечания	377
п. 3.2. Проблема выбора	379
п. 3.3. Алгоритм решения проблемы выбора	380
п. 3.4. Обоснование алгоритма решения проблемы выбора	383
п. 3.5. Общая транспортная задача, алгоритм	393
п. 3.6. Обоснование алгоритма решения общей транспортной задачи	398
п. 3.7. Пример	402
п. 3.8. Особенности венгерского метода	404
Глава 6. Математические основы линейного программирования	408
§ 1. Конечномерные пространства и выпуклые множества	408
п. 1.1. Конечномерное векторное пространство	408
п. 1.2. Выпуклые множества	415
п. 1.3. Крайние точки множества и их свойства	420
п. 1.4. Выпуклые конусы	426
§ 2. Доказательства теорем двойственности	434
§ 3. Обоснование некоторых утверждений главы 5	441
п. 3.1. Свойства планов транспортной задачи (дополнение к § 1 гл. 5)	441
п. 3.2. К обоснованию метода потенциалов (дополнение к § 2 главы 5)	444
п. 3.3. Венгерский метод как уточнение метода сокращения невязок применительно к транспортной проблеме (дополнение к § 3 главы 5)	452
Глава 7. Заключение	459
§ 1. Некоторые специальные вопросы линейного программирования	461
п. 1.1. Выпуклое программирование	461
п. 1.2. Целочисленное программирование	464
п. 1.3. Учет специфики задач	465
§ 2. Линейное программирование и теория игр	467
§ 3. Перспективные вопросы линейного программирования	477
п. 3.1. Параметрическое программирование	477
п. 3.2. Линейное программирование в условиях неопределенности	480
Литература	484

Д. Б. Юдин, Е. Г. Гольштейн

ЗАДАЧИ И МЕТОДЫ ЛИНЕЙНОГО ПРОГРАММИРОВАНИЯ

Редакторы *С. М. Мовшович,*
Н. Д. Иванушко

Техн. редактор *А. А. Свешников*
Обложка худож. *В. Т. Сидоренко*

Сдано в наб. 20.VII.60 г. Подп. к печ. 23.V.1961.

Формат 84×108/32. Объем 25,22.

Г-77189. Уч.-изд. листов 24,788

Тираж 10 000. Заказ 3/750.

Цена в переплете № 5 — 1 р. 34 к.,

№ 7 — 1 р. 39 к.

Ленинградская типография Госгортехиздата
Ленинград, ул. Салтыкова-Щедрина, 54

ЗАМЕЧЕННЫЕ ОПЕЧАТКИ

Страница	Строка	Напечатано	Должно быть
73	19 сверху	Ю. Робинсоном	Ю. Робинсон
158	4 сверху	$\sum_{i \in B_1}$	$\sum_{i \in B_1}$
189	13 сверху	0, 1, 2, 5	0, 12, 5
209	1 снизу	$z_s A_{s_l}$	$z_{s_l j} A_{s_l}$
355	2 снизу	кроме $c_{i_0 j_0}$, — X_k , существенны.	кроме $c_{i_0 j_0}$, X_k — существенные.
373	1 снизу	$x_{i_1 j_1} \triangleright x_{i_2 j_2}$	$x_{i_1 j_1} > x_{i_2 j_2}$
387	13 снизу	$(z_{i_{t+1} j_t})$	$(z_{i_{t+1} j_t})$
406	2 сверху	$\mathcal{L}_{\min}$	$L_{\min}$
445	9 снизу	$\sum_{i=1}^n v_j$	$\sum_{j=1}^n v_j$